Agosto 1984 - Lire 3.000 - Anno 1 - n° 2 - Sped. Abb. Post. Gr. III/70 - CR - Distr. MePe

COMMODORE

MENSILE PER UTENTI DI VIC 20 & COMMODORE 64

IL LIST COME LO FACCIAMO NOI

•

STAMPA FATTURA

•

TITOLATRICE

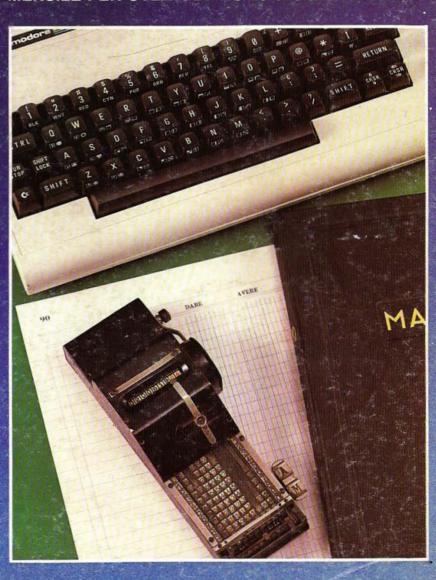
•

RISOLUZIONE DI EQUAZIONI

•

IL PROBLEMA DEI CHICCHI DI GRANDINE

•

CHARACTER EDITOR

FINALMENTE.
LA TAVOLETTA GRAFICA A PIENE PRESTAZIONI AD UN PREZZO ACCESSIBILE A TUTTI

Koala
Disponibile per Apple][+ e //e
Atari 400 e 800, Commodore 64
ed IBM P.C.

La tavoletta grafica KOALA è la più simpatica innovazione nel campo dei personal computers. Con KOALA, controllate il vostro computer con un dito. Più veloce di un paddle, più versatile di un joystick e più semplice di una tastiera.
La tavoletta grafica KOALA è compatibile con la maggior parte di software esistente e viene fornita completa del suo programma grafico "Micro Illustrator".
KOALA-PAD è il miglior modo per creare immagini ad alta risoluzione con il vostro computer.

COMPUTER GRAPHICS DIVISION
MILANO: Via L. da Vinci, 43 - 20090 Trezzano S/N
Tel. 02/4455741/2/3/4/5 - Tlx: TELINT I 312827
ROMA: Via Salaria, 1319 - 00138 Roma
Tel. 06/6917058-6919312 - Tlx: TINTRO I 614381

DIRETTORE RESPONSABILE
Agostina Ronchetti

REDATTORE CAPO
Gloriano Rossi

SEGRETARIA DI REDAZIONE
Maura Ceccaroli

GRAFICA e IMPAGINAZIONE
Renato Caruso
Francesco Amatori

FOTO
Franco Vignati

DIFFUSIONE E ABBONAMENTI
Marina

DIREZIONE, REDAZIONE
V.le Famagosta, 75
20142 Milano - Tel. 02/8466675
Autorizzazione del Tribunale di Milano
n. 103 del 25/2/84

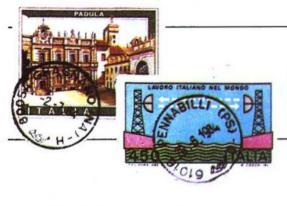
STAMPA
Lito 3 (Cologno M.)

Concessionario esclusivo
per la diffusione - MEPE spa
Via G. Carcano, 32 - Milano

Spedizione in abbonamento
postale - Gruppo III/70

Prezzo della rivista L. 3.000
Numero arretrato L. 6.000

Abbonamento annuo L. 25.000
I versamenti vanno indirizzati
a: Commodore C.C.
V.le Famagosta, 75 - 20145 Milano,
mediante emissione di assegno
bancario utilizzando il c/c postale
n.ro 31532203

Per i cambi di indirizzo, indicare, oltre naturalmente al nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500 anche in francobolli.

LA POSTA

• Stupendo il programma sulle protezioni, ma se io volessi listare o addirittura "copiare" do es. un dischetto protetto che fare? Non voglio certo mettere in commercio copie pirata, però mi farebbe comodo avere una o due copie di un programma *(pardon dischetto)* protetto e costoso.

(Luciano Frattarelli - Roma)

☐ Non esiste in realtà una protezione veramente insproteggibile. Ecco perché chi sviluppa software studia protezioni complicate a tal punto che la difficoltà di sprotezione sia direttamente proporzionale alla qualità del programma. Ecco anche perché chi sprotegge studia sistemi sempre più sofisticati per copiare. È probabile, quindi, che il suo originale abbia una protezione tale che sia impossibile eseguire un Backup dell'intero dischetto. In questi casi gli smaliziati utilizzano il programma FCOPY 4.5 ed in certi casi anche UNGUARD i quali copiano anche le tracce appositamente danneggiate e simulano le condizioni di errore volute dal programma oggetto.

• Ho acquistato un Commodore 64 circa 20 giorni fa, sono pertanto un principiante mentre la vostra rivista è per esperti super. Consiglio di inserire una parte per principianti. In caso contrario non potrò acquistare la rivista, penso che anche altri lettori abbiano lo stesso problema.

(Viali Franco - Sulmona)

☐ Gli argomenti trattati e che saranno sviluppati su Commodore sono e saranno sempre per i vari stadi di conoscenza della macchina.

Mi domando se parlassi di PRINT piuttosto che di GOSUB, per quanto tempoi avrei la sua attenzione felice possessore di C64 da 20 giorni.

Non è meglio che la spiegazione di concetti base del BASIC siano impliciti nei vari articoli o nelle REM *(commenti)* del listato?

• Possiedo un Commodore Vic 20. Nel n. 1 di Commodore voi insegnate dei tipi di protezione, alcune di queste come quella antisave non funzionano dul VIC 20. Vorrei sapere se è necessario cambiare alcuni valori es. PEEK *839).*

(Alberto Zalla — Traviso)

☐ Per disabilitare correttamente il SAVE sul Vic 20 occorre eseguire: POKE 818,73 e per riabilitare detto comando: POKE 818,113

• Sono uno studente d'ingegneria civile che utilizza un CBM 64 per calcoli strutturali; ho dei problemi di precisione numerica che necessiterebbero, per essere evitati, il passaggio in "doppia orecisione". Come si può fare?

(Enrico Albertoni - Padova)

☐ È un argomento estremamente interessante. Mi ricordo, se non erro, di avere la risoluzione sia in BASIC che in Assembler. È una promessa il suo quesito sarà tema di un prossimo articolo.

• Desidererei vedere pubblicati più programmi per il VIC 20 anche espanso 16. Grazie.

(Raineri Antonio - Melegnano - Milano)

Uso il Vic nell'insegnamento. Mi consigliate un testo sull'organizzazione didattica *(matematica)* **con tale macchina?**

(Malacrina Marco - Milano)

☐ Sia per il caro lettore Raineri che per il sig. Malacrina, posso affermare che molti programmi che vengono e che verranno pubblicati su Commodore sono di estrazione didattica matematica e anche se non specificato vanno bene anche per il Vic 20. Per altri problemi didattici posso consigliare sia i libri della nostra casa editrice che le pubblicazioni della Società Editrice Didattica Informatica con sede in Monza (Milano) Viale C. Battisti n. 44.

• Ho letto sul n. 1 di Commodore l'articolo "Flussi Relativi", a Sua firma. Dalle poche righe lette si evince che la Sua preparazione è notevole. Questo mi spinge, oltre che a complimentarmi con Lei, a chiederle se gentilmente potesse consigliarmi un ragguardevole testo circa i files relatives, oppure inviarmi, dietro compenso che vorrà preannunciarmi, delle fotocopie di dispense sulle quali Lei stesso ha basato la Sua preparazione. Nel ringraziarla anticipatamente, porgo distinti saluti.

(Maccarone Giuseppe - Roma)

☐ Caro sig. Maccarone, è inutile dire che a ringraziarla sia io. Infatti dopo le parole che mi ha dedicato mi sono alzato di almeno un palmo. Ma bando a ciò, mi premuro a cercare di darLe una risposta il più possibile valida. Non esiste in realtà una pubblicazione valida che tratti in maniera sufficientemente l'argomento inerente ai files relativi. La mia esperienza, infatti, si basa sull'esperienza acquisita nei numerosi anni sui grossi computers e poi, da sette anni circa, sui personal delle varie specie ed infine dalla vasta biblioteca sull'informatica di cui sono gelosissimo e felice possessore. Leggendo un po' qui e un po' la e poi facendo prove, ecco la risoluzione dei problemi. Quale consiglio allora? Acquistare qualche libro che parli di gestione dei files in generale, e poi personalizzare i concetti appresi ai singoli problemi.

IL LIST COME LO FACCIAMO NOI

di Gloriano Rossi (I2KH)

È oramai storia: il primo personal computer ad entrare in Italia è stato un Commodore. Si chiamava PET-2001. Ricorreva l'anno 1977. Lo acquistai subito.

Quel piccolo animalino domestico aveva incorporato: la CPU con 8 KRAM, il video, la tastiera ed il registratore. Ero abituato da anni con i grossi computer, e trovarmi con quel piccolo "coso" mi dava un entusiasmo senza pari. Nei centri meccanografici vige la legge per la quale ogni programma o procedura è catalogata, documentata in maniera tale che in ogni momento sia possibile una consultazione a tavolino. Intendo proprio la documentazione cartacea. Questo modo di lavorare mi è sempre rimasto dentro, e... appena è stato possibile ho comperato una stampante, Commodore naturalmente.

Questa obbligatorietà sul tipo di stampante da acquistare era ed è dettata dal fatto che tutti i computer Commodore utilizzano caratteri semigrafici, nonché simboli inerenti a comandi di controllo cursore che ben conosciamo. Questo fatto, sotto certi aspetti estremamente positivo, ne possiede alcuni negativi. Infatti, se io avessi acquistato un altro tipo di stampante avrei avuto una non lieta sorpresa nel momento in cui avrei incominciato a listare dei programmi. I controlli cursore ed i simboli semigrafici sarebbero stati bellamente ignorati, ma non solo, non sarebbero stati nemmeno interpretati o tradotti con spazi od altro, ma invece solamente considerati inesistenti, e quindi nessuna stampa viene eseguita. Ciò naturalmente porta ad un notevole inconveniente nella lettura di un programma scritto su carta da una stampante NO-Commodore.

Circa nel 1979, trovai su una rivista (inglese o americana, non ricordo bene) un articoletto che tentava di risolvere l'annoso problema.

L'autore suggerriva di digitare, in coda ad ogni programma da listare, una decina di righe che servivano, appunto a decodificare, direttamente dalla memoria, gli eventuali simboli di controllo cursore, l'HOME, il CLR, il DOWN, l'UP. Tutto ciò aveva molti inconvenienti e quindi, nonostante possedessi una stampante adeguata, mi sono accinto ad ampliare e modificare sempre più quell'idea di massima che senza alcun dubbio poteva essere valida.

Sono passati alcuni anni, e nel frattempo il programma LIST, ha subito moltissime implementazioni ed è stato "regalato", come parecchi altri programmi/utilities che ho fatto in passato, a moltissimi possessori di Commodore.

L'ultima edizione, quella che vi presento, ha subito un'ulteriore modifica, quella sostanziale che mi permette di poter leggere il programma da listare direttamente da disco.

Il programma

Nella prima parte del LIST, quella cioè che va **dalla riga 1110 alla riga 1240**, si definiscono tutti i parametri fissi relativi al tipo di stampante ed al tipo di stampa. Infatti è possibile eseguire la stampa sia con una stampante Commodore che con una stampante NO Commodore. Nel primo caso avremo la decodifica di tutti i comandi speciali, mentre i caratteri grafici vengono riportati integralmente.

Nel secondo caso, quando cioè, la stampante in uso non prevede in alcun modo le simbologie speciali, come ad esempio le stampanti a margherita, avviene sempre la traduzione dei comandi, mentre i caratteri grafici vengono interpretati con un semplice punto.

È quindi logico che la prima domanda fosse inerente proprio alla conoscenza del tipo di stampante **(righe 1120-1160)**. La risposta dovrà essere o 0 o 1, qualsiasi carattere diverso viene ignorato.

Segue a questo punto un periodo, breve naturalmente, di attesa, per permettere al computer di immagazzinare nella tabella (array) tutte le parole dell'interprete BASIC poste nel giusto ordine secondo il codice di richiamo. **Riga 1170 e routine da 1560-1770**.

Sappiamo infatti che una parola BASIC, in un programma occupa un

commodore
COMPUTER
CLUB

Presenta

Commodore
Club
IN CASSETTA
N. 1
Lire 4.800
NOVITÁ

solo byte in memoria, ed il contenuto di questo byte corrisponde al codice numerico che in ogni caso è sempre superiore al numero 127. Sapete tutti infatti che ad esempio la parola GOBUS non occupa in memoria 5 caratteri, ma bensì un solo carattere il cui contenuto è 141. Sarà poi la routine del BASIC di LIST oppure, nel nostro caso il programma LIST, ad effettuare il giusto abbinamento fra codice e l'effettiva parola BASIC.

Alla domanda se si vuole l'impaginazione oppure no, si potrà rispondere esclusivamente affermativamente (s) o negativamente (n), in caso contrario ogni carattere verrà ignorato. **Righe 1190-1210.**

Ad una risposta negativa, quella che io utilizzo quando devo listare i programmi per la rivista, si ottiene una stampa senza il salto pagina. **Righe 1220-1230.** Se invece rispondo affermativamente, quando eseguo i listati per i libri, il programma chiede anche il formato fisico della carta. Occorre cioè far sapere al computer quante siano le righe del foglio. Sarà il programma stesso che provvederà a stampare per ogni pagina l'intestazione con la relativa numerazione, tranne per la prima pagina, e provvedere ad eseguire tanti salti riga per poter saltare la zinigratura del modulo continuo.

Un dato estremamente importante da far conoscere al computer è quello relativo al numero massimo di caratteri per ogni riga di tabulato che deve essere stampato. **Riga 1240.**

L'input del nome del programma su disco deve essere dato con precisione, proprio come è scritto sulla directory e come è stato salvato in precedenza. **Riga 1250.**

Invece per l'intestazione del tabulato può essere data una serie di caratteri e parole a piacere facendo attenzione di non superare i trenta/trentacinque caratteri. **Riga 1260.**

Se alla variabilità CM abbiamo dato un valore inferiore a 41 è apprezzabile vedere il risultato del programma, anche su video. Ecco il perché della richiesta Video o Stampante. Il BASIC del Commodore ci permette di aprire un file di output, tipo stampa, su vari tipi di device.

numero	device	tipo
0	tastiera	input
1	registratore	input/output
2	modem	input/output
3	video	output
4	stampante	output
5	stampante	output
6	plotter	output
7-8-9	dischi	input/output

Dopo aver fornito tutti questi parametri, inizia la parte vera e propria di esame del programma da listare. **Riga 1300.** Noi possiamo aprire un file, sia in input, come nel nostro caso, che in output, nel formato programma, infatti la OPEN che ho utilizzato prevede proprio questo tipo di apertura (PRG, Read).

Il BASIC quando salva un programma (SAVE) prima di scriverlo memorizza in due bytes iniziali l'indirizzo di partenza della memoria RAM riferentesi al programma oggetto. Nel nostro programma però non esiste interesse per questa informazione, quindi si leggono due bytes a vuoto. **Riga 1310.**

Definiamo ora le variabili di OPEN di output. **Riga 1320.**

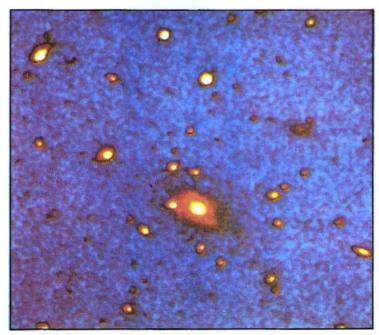
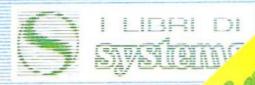
Ed eseguiamo la OPEN relativa. **Riga 1330.**

A questo punto si presuppone che sia stata allineata la carta, ed infatti il computer provvede ad eseguire la prima intestazione, la quale non riporterà il numero di pagina solo per un fatto estetico. **Riga 1340.**

Infine setta le variabili di contarighe e contapagine con i dovuti valori. **Riga 1350.**

Ora inizia l'esame del programma, riga per riga.

Leggo quattro bytes. **Riga 1360.**

I primi due mi dicono quale sarebbe la locazione della momoria RAM

dove inizierebbe la prossima riga BASIC.

I secondi due bytes mi forniscono il numero di riga attuale nell'ambito del programma oggetto. È inutile dire che la numerazione di riga, sia che abbia un carattere che ne abbia cinque, occupa sempre due bytes, perché detto numero viene espresso e memorizzato in esadecimale. Ricordate infatti che la numerazione di riga BASIC può andare da 0 a 65535 (esadec. 0000 e FFFF). **Righe 1370-1410.**

Il BASIC definisce l'ultimo byte di ogni riga a 00, proprio per riconoscere la fine della riga, e dopo l'ultima riga pone sempre a OO ulteriori due bytes. Ecco perché se il puntatore alla prossima riga è uguale a zero, il programma è finito e quindi si termina il listato e si ritorna alla richiesta di un altro programma da listare. **Riga 1420.**

Iniziamo con una costruzione della stringa contenete la riga BASIC. La prima informazione è naturalmente relativa al numero di riga posto in esatta giustificazione (sei caratteri, cinque per il numero più uno spazio). **Riga 1440.**

Leggiamo un byte. Se questo ha un valore valido ne prendiamo l'esatto valore ASCII. **Riga 1450.**

Se il carattere è zero, la riga è finita. Se ciò fosse vero, controllo se è stata aperta una parentesi quadra (GOSUB 2320). Provvedo a stampare la mia riga (GOSUB 2520) e infine vado ad esaminare una nuova serie di bytes. **Riga 1460.**

Se la variabile Q ha un contenuto e il carattere in P è un doppio apice, vuole dire che quest'ultimo corrisponde ad una chiusura di stringa e quindi occorre verificare se è stata aperta una parentesi quadra (GOSUB 2320). **Riga 1470.**

Se il carattere in esame è un doppio apice allora vario il valore in Q. Infatti se Q esiste, lo cancello, al contrario se Q non esiste lo faccio vivere (Q= NOT Q). **Riga 1480.**

Se Q esiste, quindi siamo in apertura di apici, eseguo la routine di riconoscimento comandi (GOSUB 1840). **Riga 1490.**

Se siamo fuori apici ed il valore in P è superiore a 127, tranne per il p-greco (carattere 255), siamo in presenza di un codice di parola BASIC, quindi la identifico in tabella ed eseguo il concatenamento con la stringa XT$. **Righe 1500-1510.**

Invece se non è una parola BASIC è un nome di variabile o un segno operativo, quindi lo considero come un carattere singolo e lo concateno con XT$. **Riga 1520.**

Se siamo fuori apici devo sapere se il comando BASIC equivale ad una REM. In caso affermativo in seguito mi devo comportare, per i prossimi caratteri, come se fossero stati aperti i doppi apici. **Riga 1530.**

A questo punto il programma LIST può considerarsi finito in quanto le istruzioni seguenti sono relative alle routines richiamate precedentemente.

Le subroutines

Parole BASIC e caricamento in T$

Questa routine prevede la definizione dei DATA con tutte le parole BASIC del Commodore 64 e di tutti gli altri titpi di computer Commodore, infatti sono presenti anche le parole del BASIC 4.0. È molto importante che si rispetti tassativamente la sequenza con la quale sono state riportate con i dovuti spazi che rendono il listato più leggibile.

Infatti il Commodore ci pemette di scrivere un programma in BASIC senza tenere conto delle spaziature fra istruzioni e variabili, fatto questo obbligatorio per altri tipi di BASSIC tipo l'MS-BASIC o il BASIC-86. Questo naturalmente va a discapito della leggibilità del listato.

Gestione dei caratteri fra apici

Controllo se il valore ASCII del carattere attuale corrisponde ad un comando. In caso affermativo, verifico se è uguale al prcedente (GOSUB 2320). Memorizzo l'attuale valore. Incremento il contatore del comando in P e ne attribuisco una stringa il cui contenuto è la spiegazione del comando stesso. **Righe 1840-2210.**

Se il carattere attuale è diverso dal precedente e il contatore di P è diverso da zero, vuol dire che è terminato un carattere di comando o serie dello stesso e quindi occorre completare la stringa CS$. **Riga 2210.**

Se ho dichiarato che il tipo di stampante non è Commodore, quindi la variabile TS è superiore a zero, e siamo in presenza di un carattere grafico è necessario trasformare questo carattere in uno previsto da tutti i tipi di printers, ovvero il punto, infatti il codice 46 corrisponde al punto.

Se invece il codice è un carattere ASCII normale esiste una normale trascodifica dello stesso. **Righe 2220-2240.**

Creazione di stringa in parentesi quadre

Se il carattere attuale è uguale al precedente oppure se il contatore di P è a zero allora non devo fare nulla e quindi ritorno. **Righe 2320-2330.** Costruisco una stringa CP$ contenente il numero di vlte che è stato riconosciuto il comando e se questo corrisponde ad una sola volta, allora detta stringa è annullata. **Riga 2340.**

Il codice espresso in tringa si trasforma ora nel formato richiesto fra

parentesi quadre. Eseguo il concatenamento e modifico opportunamente le variabili di controllo. **Righe 2350-2380.**

La stringa attuale corrisponde alla precedente stringa attuale con in aggiunta la nuova stringa che potrà contenere un carattere, un comando speciale o una parola BASIC. **Riga 2440.**

Stampa la riga

la riga BASIC in esame è completa ed è presente nella variabile XT$. MI informo sulla sua lunghezza. **Riga 2520.**

Se questa lunghezza è minore o uguale al numero di caratteri massimo per riga di tabulato, allora posso tranquillamente stampare senza fare alcuna modifica. . **Riga 2530.**

Devo stampare quella parte (sinistra) di XT$ con lunghezza richiesta. **Riga 2540.**

Il nuovo contenuto di XT$ corrisponderà al precedente contenuto al quale è stato tolto ciò che è presente in X$ (variabile da stampare) preceduto da sei spazi per l'incolonnamento. **Riga 2550.**

Se avevo richiesta l'impaginazione la variabile PP% è senza alcun dubbio maggiore di zero e quindi provvedo ad eseguire la subroutine di impaginazione. **Riga 2570.**

Stampo X$. **Riga 2560.**

Se il contenuto della variabile transitoria X$ corrisponde a quello contenuto dalla XT$ vuole dire che ho terminato di stampare la riga BASIC esaminata, altrimenti ripeto il ciclo di stampa. **Righe 2580-2590.**

Gestione dell'impaginazione

Incremento il contatore delle righe stampate. **Riga 2660.**

Se il contatore di righe stampate è inferiore o uguale al numero di righe stampabili allora posso ritornare. **Riga 2670.**

Altrimenti eseguo cinque stampe a vuoto per saltare la divisione del foglio e quindi ripeto l'intestazione con la numerazione di pagina, salto quindi un'altra riga ed infine, prima di tornare dalla subroutine, setto opportunamente le variabili di conteggio riga e pagina. **Righe 2680-2710.**

Pricipali variabili utilizzate

- TS Tipo stampante (0 = Commodore, 1 = non Commodore).
- PP% Formato modulo continuo, espresso in numero righe.
- PG% Righe stampabili.
- CM Numero dei caratteri per riga.
- NP$ Nome Programma, così come è memorizzato sul disco.
- VS$ Nome del Programma da riportare nell'intestazione del tabulato.
- NN Numero interno di file relativo all'output.
- ND Numero di device di output (4 = stampante, 3 = video).
- CR% Contatore di righe stampate.
- CP% Contatore di pagine stampate.
- L Numero della riga BASIC in esame.
- P$ Byte attuale della riga in esame.
- Q Flag di apertura e chiusura dei doppi apici o REM.
- T$(Tabella delle parole BASIC.
- MP Memoria di P, in caso in cui P sia un comando speciale.
- CP Contatore di P in caso che il comando speciale sia multiplo.
- CP$ Stringa della variabile CP.
- CS$ Sviluppo in stringa del comando speciale.
- XX$ Carattere attuale o stringa del comando speciale.
- XT$ Stringa completa di tutti gli XX$ della riga attuale.
- LX Lunghezza della stringa da stampare.
- X$ Variabile di comodo.

Conclusione

È bene precisare che l'esecuzione di LIST in forma BASIC interprete può risultare alquanto lenta. È quindi consigliabile compilare il programma o con il PET-SPEED o con L'AUSTROCOMPILER od un altro compilatore BASIC. I risultati, in termine di tempo, con il compilato sono tutta un'altra cosa.

Chi volesse avere sia il listato in BASIC che il compilato, può rivolgersi alla redazione o direttamente all'autore.

```
1000 REM ***************************
1010 REM *   ROUTINE DI   L I S T  *
1020 REM ***************************
1030 REM *  CARICARE LIST E QUINDI *
1040 REM * LISTARE UN PROGRAMMA SU *
1050 REM *         SU DISCO        *
1060 REM ***************************
1070 REM * AUTHOR SOFTWARE         *
```

```
1080 REM *      I2KH GLORIANO ROSSI *
1090 REM ***********************KH**
1100 :
1110 PRINT"[CLEAR][3 RIGHT][RVS]  GLORIANO ROSSI I2KH  "
1120 PRINT"[3 DOWN][3 RIGHT][RVS]  TIPO DI STAMPANTE  "
1130 PRINT"[3 DOWN][5 RIGHT]0  COMMODORE   0  "
1140 PRINT"[3 DOWN][5 RIGHT]1  ALTRI MIDELLI[2 DOWN]"
1150 GET X$:IF X$="0" THEN 1170
1160 IF X$<>"1" THEN 1150
1170 PRINT TAB(20)X$:PRINT TAB(10)"[3 DOWN][RVS] ATTENDERE PREGO ":GO
     SUB 1610
1180 TS=VAL(X$):PRINT"[CLEAR]"
1190 PRINT"VUOI L'IMPAGINAZIONE (S/N)"
1200 GET X$:IF X$="N" THEN PP%=0:GOTO 1240
1210 IF X$<>"S" THEN 1200
1220 INPUT "FORMATO PAGINA   66[4 LEFT]";PP%
1230 PG%=PP%-4
1240 INPUT "[2 DOWN]CARATTERI PER RIGA ";CM
1250 INPUT "[2 DOWN]NOME PROGRAMMA SU DISCO    ";NP$
1260 INPUT "[2 DOWN]INTESTAZIONE SU TABULATO  ";PG$
1270 INPUT "SU VIDEO O STAMPANTE (V/S) S[3 LEFT]";VS$
1280 IF VS$="V" OR VS$="S" THEN 1300
1290 GOTO 1270
1300 OPEN 2,8,2,"0:"+NP$+",PRG,R"
1310 GET #2,A$,A$
1320 NN=4:ND=4:IF VS$="V" THEN ND=3
1330 OPEN NN,ND
1340 PRINT#NN:PRINT#NN:PRINT#NN,PG$:PRINT#NN
1350 CR%=4:CP%=2:REM  CONTARIGHE,CONTAPAGINE
1360 GET #2,A$,B$,C$,D$
1370 A=0:IF A$<>"" THEN A=ASC(A$)
1380 B=0:IF B$<>"" THEN B=ASC(B$)
1390 C=0:IF C$<>"" THEN C=ASC(C$)
1400 D=0:IF D$<>"" THEN D=ASC(D$)
1410 L=C+256*D:X=A+256*B:Q=0
1420 IF X=0 THEN FOR I=CR% TO PP%:PRINT#NN:NEXT:CLOSE NN:CLOSE 2:GOTO
      1250
1430 IF L<F THEN A=X:GOTO 1410
1440 XT$=RIGHT$("      "+STR$(L),5)+" "
1450 GET #2,P$:P=0:IF P$<>"" THEN P=ASC(P$)
1460 IF P=0 THEN GOSUB 2320:GOSUB 2520:A=X:GOTO 1360
1470 IF Q AND P=34 THEN GOSUB 2320
1480 IF P=34 THEN Q= NOT Q
1490 IF Q THEN GOSUB 1840:GOTO 1450
1500 IF P=255 THEN 1520
1510 IF  NOT Q AND P>127 THEN XX$=T$(P-127):GOSUB 2440:GOSUB 1530:GOT
     O 1450
1520 XX$=CHR$(P):GOSUB 2440:GOTO 1450
1530 IF  NOT Q THEN IF P=143 THEN Q= NOT Q
1540 RETURN
1550 :
1560 REM **************************
1570 REM * PAROLE BASIC + ROUTINE  *
1580 REM * CARICAMENTO IN ARRAY T$ *
1590 REM **************************
1600 :
```

```
1610 DATA "END ","FOR ","NEXT","DATA ","INPUT#","INPUT ","DIM "
1620 DATA "READ ","LET ","GOTO "," RUN","IF ","RESTORE ","GOSUB "
1630 DATA "RETURN","REM ","STOP","ON ","WAIT ","LOAD ","SAVE "
1640 DATA "VERIFY ","DEF ","POKE ","PRINT#","PRINT"," CONT"
1650 DATA "LIST ","CLR ","CMD ","SYS","OPEN ","CLOSE ","GET "
1660 DATA "NEW "," TAB("," TO ","FN","SPC("," THEN "," NOT "," STEP "
1670 DATA "+","-","*","/","↑"," AND "," OR ",">","=","<"," SGN"
1680 DATA "INT","ABS","USR","FRE","POS","SQR","RND","LOG","EXP"
1690 DATA "COS","SIN","TAN","ATN","PEEK","LEN","STR$","VAL"
1700 DATA "ASC","CHR$","LEFT$","RIGHT$","MID$"," GO "," CONCAT "
1710 DATA " DOPEN"," DCLOSE"," RECORD"," HEADER "," COLLECT"," BACKUP
 "
1720 DATA " COPY "," APPEND "," DSAVE "," DLOAD "," CATALOG "
1730 DATA " RENAME "," SCRATCH "," DIRECTORY "
1740 DATA  ***
1750 DIM T$(91):K=1
1760 READ T$:IF T$="***" THEN RETURN
1770 T$(K)=T$:K=K+1:GOTO 1760
1780 :
1790 REM ***************************
1800 REM *    ROUTINE DI GESTIONE    *
1810 REM *    CARATTERI FRA APICI    *
1820 REM ***************************
1830 :
1840 IF P=005 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="BIANCO":RETURN
1850 IF P=014 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="MIN-MAI":RETURN
1860 IF P=017 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="DOWN":RETURN
1870 IF P=018 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="RVS":RETURN
1880 IF P=019 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="HOME":RETURN
1890 IF P=020 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="DEL":RETURN
1900 IF P=028 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="ROSSO":RETURN
1910 IF P=029 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="RIGHT":RETURN
1920 IF P=030 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="VERDE":RETURN
1930 IF P=031 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="BLEU":RETURN
1940 IF P=129 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="ARANC":RETURN
1950 IF P=133 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="TF1":RETURN
1960 IF P=134 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="TF3":RETURN
1970 IF P=135 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="TF5":RETURN
1980 IF P=136 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="TF7":RETURN
1990 IF P=137 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="TF2":RETURN
2000 IF P=138 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="TF4":RETURN
2010 IF P=139 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="TF6":RETURN
2020 IF P=140 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="TF8":RETURN
2030 IF P=142 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="MAIUSC.":RETURN
2040 IF P=144 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="NERO":RETURN
2050 IF P=145 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="UP":RETURN
2060 IF P=146 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="RVOFF":RETURN
2070 IF P=147 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="CLEAR":RETURN
2080 IF P=148 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="INST":RETURN
2090 IF P=149 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="MARR":RETURN
2100 IF P=150 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="ROSA":RETURN
2110 IF P=151 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="GRIGIO1":RETURN
2120 IF P=152 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="GRIGIO2":RETURN
2130 IF P=153 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="VERDE2":RETURN
2140 IF P=154 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="CELESTE":RETURN
2150 IF P=155 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="GRIGIO3":RETURN
```

```
2160 IF P=156 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="VIOLA":RETURN
2170 IF P=157 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="LEFT":RETURN
2180 IF P=158 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="GIALLO":RETURN
2190 IF P=159 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="AZZUR":RETURN
2200 IF P=255 THEN GOSUB 2320:MP=P:CP=CP+1:CS$="PI":RETURN
2210 IF (P<>MP) AND CP THEN GOSUB 2320
2220 IF TS AND P>127 THEN P=46
2230 XX$=CHR$(P):GOSUB 2440
2240 RETURN
2250 :
2260 REM ***************************
2270 REM * STAMPA IN PARENTESI [ ] *
2280 REM * LE FUNZIONI DEI CARATTE-*
2290 REM * RI DI CONTROLLO CURSORE *
2300 REM ***************************
2310 :
2320 IF P=MP THEN RETURN
2330 IF CP=0 THEN RETURN
2340 CP$=RIGHT$(STR$(CP)+" ",LEN(STR$(CP))): IF CP<2 THEN CP$=""
2350 CS$="["+CP$+CS$+"]"
2360 XX$=CS$:GOSUB 2440
2370 MP=P:CS$="":CP=0
2380 RETURN
2390 :
2400 REM ***************************
2410 REM * CONCATENA RIGA IN CORSO *
2420 REM ***************************
2430 :
2440 XT$=XT$+XX$:RETURN
2450 :
2460 REM ***************************
2470 REM *  SUBROUTINE DI STAMPA    *
2480 REM *  SECONDO LA LUNGHEZZA    *
2490 REM *     RICHIESTA  (CM)      *
2500 REM ***************************
2510 :
2520 LX=LEN(XT$)
2530 IF LX<=CM THEN X$=XT$:GOTO 2560
2540 X$=LEFT$(XT$,CM)
2550 XT$="     "+MID$(XT$,CM+1)
2560 PRINT#NN,X$
2570 IF PP%>0 THEN GOSUB 2660
2580 IF X$=XT$ THEN RETURN
2590 GOTO 2520
2600 :
2610 REM ***************************
2620 REM * GESTIONE DELLA          *
2630 REM *  IMPAGINAZIONE          *
2640 REM ***************************
2650 :
2660 CR%=CR%+1
2670 IF CR%<=PG% THEN RETURN
2680 PRINT#NN:PRINT#NN:PRINT#NN:PRINT#NN
2690 PRINT#NN:PRINT#NN,PG$ TAB(40-LEN(PG$))CP%:PRINT#NN
2700 CR%=4:CP%=CP%+1
2710 RETURN
```

LA TIGRE È IN AGGUATO

State cercando una stampante per il vostro micro:

Deve essere facile da usare (manuale in italiano, selezione dei parametri da pannello e memorizzazione permanente).

Deve essere multifunzione e permettervi di passare dalla qualità listing (180 cps.) alla qualità lettera per il trattamento testi.

Deve essere facilmente interfacciabile ed immediatamente compatibile con il vostro micro... qualunque esso sia.

Deve essere lo strumento per riprodurre in modo perfetto i vostri grafici.

Deve essere molto affidabile, avere una probabilità di guasto solo ogni 18 mesi ed essere ciononostante supportata da una rete nazionale di assistenza postvendita.

Deve far parte di una gamma completa e compatibile (80 - 132 colonne, grafica, colore, inserimento del foglio singolo manuale e automatico, caratteri scientifici e APL...).

Deve sempre inserirsi nei vostri limiti di spesa e soddisfare le vostre esigenze odierne e future.

Deve essere pensata, messa a punto, prodotta e commercializzata dal PIÙ GRANDE COSTRUTTORE MONDIALE INDIPENDENTE DI STAMPANTI.

LA VOSTRA SCELTA È FATTA

100% COMPATIBILE PC IBM

SERIE SPG 8000 "PAPER TIGER"

DATAPRODUCTS s.r.l.
Via Vincenzo Monti, 8 - 20123 MILANO - Tel. 3452211-860347

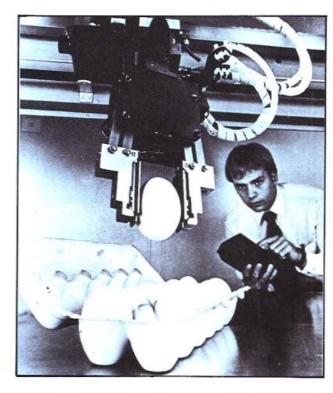
STAMPA FATTURA

di Gloriano Rossi & Giulio Marcozzi

Ecco un breve programmino di stampa fattura. Anche se molto semplice, può essere un valido strumento di lavoro per chi non abbia la necessità di una gestione computerizzata della propria contabilità oppure che non voglia far stampare le proprie fatture dal tipografo ecc. Un programma si presta tra l'altro indubbiamente a molteplici implementazioni: dall'aggancio con un archivio dei clienti, alla memorizzazione degli importi e... chi più ne ha più ne metta.

Il programma

La routine iniziale è stata messa in coda al programma volutamente, in quanto viene eseguita una sola volta alla partenza e non viene più interessata da ulteriori GOSUB. Segue poi la stampa grafica della propria ragione sociale. È chiaro che ciò che viene riportato nel listato e quindi anche nell'esempio deve essere personalizzato secondo le proprie esigenze o le esigenze dell'amico o cliente al quale sarà destinato. Sarà allora necessario effettuare un po' di prove con la stampante, nel nostro esempio la MPS801, al fine di ottenere un effetto grafico accettabile.

Nell'esempio è riportato il mio nominativo di radioamatore (i2KH), per far capire come si possano costruire dei caratteri in maniera grafica. Tutti i dati del cliente vengono stampati in maniera e posizione tale da poter sfruttare quelle buste con finestra che sono normalmente in commercio. Il programma prevede la possibilità di stampare più copie della medesima fattura riportando, in fondo, l'identificazione di quale copia si tratti. Se la dicitura non fosse di gradimento è sempre possibile modificarla intervenendo sulle righe 317-320.

La quantità di copie di default corrisponde a tre, ma se si utilizza un modulo continuo multicopia sarà sufficiente azzerare gli array CO$ (e porre a 1 il contatore di copie (variabile NC).

La prima copia di fattura viene stampata mano a mano che viene costruita, mentre le successive sono stampate in una sola volta; infatti ogni riga viene memorizzata nell'array RI$(.

Tutto il listato è commentato sufficientemente con delle REM in maniera tale che sia possibile individuare le varie routines e quindi più facilmente è possibile intervenire con modifiche e personalizzazioni.

STAMPA FATTURE

```
100 REM   ************************
101 REM   ************************
102 REM   **                    **
103 REM   **   STAMPA FATTURE   **
104 REM   **                    **
105 REM   **         DI         **
106 REM   **                    **
107 REM   **   GLORIANO ROSSI   **
108 REM   **                    **
109 REM   **  GIULIO  MARCOZZI  **
110 REM   **                    **
111 REM   ************************
112 REM   ************************
113 GOSUB 311
114 Z=0
115 CN=1:REM   CONTACOPIE
116 OPEN 4,4
117 NF$=AA$+"[UP]F[DOWN]ATTURA "+
    STR$(NF)+NN$
118 REM   ************************
119 REM   * STAMPA INTESTAZIONE  *
120 REM   ************************
121 PRINT#4,NN$" "MM$
122 PRINT#4,NN$" "MM$
123 PRINT#4,NN$"       ▪     [RVS] ▪[
    RVOFF][RVS]▬[RVOFF][RVS]▪ [RVO
    FF]    [RVS]▌[RVOFF]  [RVS]▚[
    RVOFF]   ▌ [RVS]▌[RVOFF]"MM$
124 PRINT#4,NN$"       ▌     ▌
    [RVS]▌[RVOFF] ▪[RVS] [RVOFF]
     ▌ [RVS]▌[RVOFF]           "MM
    $
125 PRINT#4,NN$"      [RVS] I[RVOFF
    ] ▬▬ [RVS] ▪[RVOFF]\[RVS]▪[RVO
    FF] ▬▬ [RVS]▌[RVOFF]▌ [RVS]▪[
    RVOFF]   [RVS] [RVOFF]▔[RVS]▪
    [RVOFF]
126 PRINT#4,NN$"      [RVS] I[RVOFF
    ]    ▌       [RVS]▌[RVOFF]▌ [R
    VS]▌[RVOFF]   [RVS] [RVOFF] [
    RVS]▌[RVOFF] "MM$
127 PRINT#4,NN$"      [RVS] I[RVOFF
    ]    [RVS]▔▔[RVOFF]     [RVS]
    ▌[RVOFF]▌ [RVS]▌[RVOFF]    [RV
    S] [RVOFF] [RVS]▌[RVOFF]"MM$
128 PRINT#4,NN$" "MM$
129 PRINT#4,NN$" "MM$
```

```
130 PRINT#4,NN$"     [UP]S[DOWN].P
    .A DI [UP]P[DOWN]IPPOLI  [UP]
    P[DOWN]IPPO       "SPC(20)NF$
131 PRINT#4,"    [UP]V[DOWN]IA  [
    UP]P[DOWN]IPPONIO N.46 - 2012
    1 [UP]M[DOWN]ILANO"
132 PRINT#4,"    T[DOWN]EL.(02) 9
    9.99.999 - 66.66.666  T[DOWN]
    ELEX N.444444"
133 PRINT#4,"    C.F.[DOWN]E [UP]
    P I.V.A.[DOWN] N.1234567890 "
134 FOR I=1 TO 3
135 PRINT#4
136 NEXTI
137 IF CN=1 THEN PRINT"[CLEAR]":G
    OSUB 279:REM  INPUT DATI CLIE
    NTE
138 PRINT#4, TAB(46)"S[DOWN]PETT.
    [UP] "
139 PRINT#4, TAB(46)A$
140 PRINT#4, TAB(46)B$
141 IF  CN=1 THEN  C$=E$+" - "+C$
    +" ":IF LEN(D$)>0 THEN C$=C$+
    "("+D$+")"
142 PRINT#4, TAB(46)C$
143 PRINT#4
144 PRINT#4
145 PRINT#4,"    [UP]M[DOWN]ILANO
     : ";F$
146 PRINT#4,"    [UP]R[DOWN]IFERI
    MENTI ";RF$
147 PRINT#4,"    [UP]V[DOWN]OSTRO
     [UP]C.F[DOWN] E/O[UP] P.I[DO
    WN]VA N. [UP]";CF$
148 PRINT#4,"       ";
149 PRINT#4,R1$;
150 PRINT#4, TAB(5)ZZ$
151 PRINT#4,"       ";
152 PRINT#4,R1$;
153 PRINT#4
154 REM  ************************
155 REM  *  INPUT DATI FATTURA  *
156 REM  ************************
157 IF  CN>1 THEN 191
158 PRINT"[CLEAR][2 DOWN]DESCRIZI
    ONE ARTICOLO"
159 PRINT"(RETURN PER FINIRE)"
160 INPUT " 0[3 LEFT]";G$
161 IF LEN(G$)>35 THEN QU$=LEFT$(
    G$,35):G$=QU$:QU$=""
162 IF G$="0" THEN 198
163 QU=0
164 PRINT
165 INPUT "N' PEZZI ";QU
166 QU$=STR$(QU)
167 B=0
168 PRINT
169 INPUT "PREZZO CADAUNO ";B
170 PRINT
171 PRINT
172 INPUT "CONFERMA ( 0=SI * 1=NO
     )     0[3 LEFT]";CO
173 IF CO<>0 THEN PRINT"[CLEAR]":
    GOTO 158
174 PRINT"[CLEAR]"
175 VA=B
176 GOSUB 258
177 PRINT"[CLEAR]"
178 RI$(IC)="     "+G$+LEFT$(BL$,4
    0-LEN(G$))+" "+LEFT$(BL$,11-L
    EN(VA$))+VA$+" "
179 K=B*QU
180 Z=Z+K
181 RI$(IC)=RI$(IC)+LEFT$(BL$,6-L
    EN(QU$))+STR$(QU)+" "
182 VA=K
183 GOSUB 258
184 RI$(IC)=RI$(IC)+LEFT$(BL$,12-
    LEN(VA$))+VA$
185 PRINT#4,RI$(IC)
186 IC=IC+1
187 GOTO 157
188 REM  ************************
189 REM  *STAMPA RIGHE COSTRUITE*
190 REM  ************************
191 I=27-IC-1
192 FOR IC=0 TO I
193 PRINT#4,RI$(IC)
194 NEXT
195 REM  ************************
196 REM  * STAMPA RIGHE BIANCHE *
197 REM  ************************
198 IC=27-IC
199 PRINT"[CLEAR]"
200 FOR I=1 TO IC
201 PRINT#4
202 NEXTI
203 PRINT#4,"       ";
204 PRINT#4,LEFT$(R1$,60);
205 PRINT#4,"____________________"
206 PRINT#4,"       T[DOWN]OT.[UP]I[
    DOWN]MPONIB.[UP] | S[DOWN]C.[U
    P] % | T[DOWN]OT. [UP]S[DOWN]C
    ONT.[UP] | IVA % |";
207 PRINT#4,"  T[DOWN]OT. [UP]IVA
     | T[DOWN]OT[UP]  |
```

```
     |"
208 VA=Z
209 GOSUB 258
210 PRINT#4, TAB(17-LEN(VA$))VA$;
    "| ";
211 VA$=STR$(SC)
212 PRINT#4, TAB(4-LEN(VA$))VA$;"
     |";
213 VA=INT(Z-SC*Z/100+0.99)
214 GOSUB 258
215 PRINT#4, TAB(12-LEN(VA$))VA$;
    " | ";
216 VA$=STR$(IV)
217 PRINT#4, TAB(4-LEN(VA$))VA$;"
     |";
218 VA=INT(((Z-SC*Z/100)*IV/100)+
    0.99)
219 GOSUB 258
220 TV=1+IV/100
221 PRINT#4, TAB(12-LEN(VA$))VA$;
    " |F[DOWN]ATT.[UP] |";
222 VA=INT((Z-SC*Z/100)*TV+0.99)
223 GOSUB 258
224 PRINT#4, TAB(11-LEN(VA$))VA$;
    " |"
225 PRINT#4,"           __________|____
    _____|_____________|________|_____
    __________|";
226 PRINT#4,"_________|______________
    |";
227 PRINT#4, TAB(68)"S.E.& O."
228 PRINT#4,"     M[DOWN]ODALITA'
    DI PAGAMENTO - ";WW$
229 PRINT#4
230 CLOSE 4
231 IF  NC>CN THEN CN=CN+1:NV=7:G
    OSUB 250:GOTO 116
232 Z=0
233 NF=NF+1
234 PRINT"[HOME][CLEAR]"
235 PRINT"VUOI CONTINUARE ?"
236 GET R$
237 IF R$="" THEN 236
238 IF R$="S" THEN CN=CN+1:GOTO 2
    44
239 IF R$<>"N" THEN 236
240 CN=CN+1
241 NV=43
242 GOSUB  250
243 END
244 NV=7
245 GOSUB  250
246 GOTO 114
247 REM  ***********************
248 REM  *   CHIUDE LA PAGINA   *
249 REM  ***********************
250 OPEN 4,4
251 FOR I=1 TO NV
252 IF I=3 THEN PRINT#4,CO$(CN-1)
    :GOTO 254
253 PRINT#4
254 NEXT
255 CLOSE 4
256 RETURN
257 REM  ***********************
258 REM  *METTE PUNTI ALLE CIFRE*
259 REM  ***********************
260 VA$=STR$(VA)
261 VB$=RIGHT$(VA$,3)
262 VA=INT(VA/1000)
263 VA$=STR$(VA)
264 IF VA=0 THEN VC$=VB$:GOTO 274
265 VC$="."+VB$
266 VB$=RIGHT$(VA$,3)
267 IF VB$="" THEN 274
268 VA=INT(VA/1000)
269 VA$=STR$(VA)
270 IF VA=0 THEN VC$=VB$+VC$:GOTO
     274
271 VC$="."+VB$+VC$
272 VA$=STR$(VA)
273 VC$=VA$+VC$
274 VA$=VC$
275 RETURN
276 REM  ***********************
277 REM  *  INPUT DATI CLIENTE  *
278 REM  ***********************
279 PRINT"[CLEAR]"
280 INPUT "NOME CLIENTE ";A$
281 CF$=""
282 PRINT
283 INPUT "COD.FISC.CLIENTE ";CF$
284 PRINT
285 INPUT "INDIRIZZO ";B$
286 PRINT
287 INPUT "CODICE POSTALE ";E$
288 PRINT
289 INPUT "CITTA' ";C$
290 D$=""
291 PRINT
292 INPUT "PROVINCIA ";D$
293 IV=18
294 PRINT
295 INPUT "PERCENTUALE IVA ";IV
296 SC=0
297 PRINT
```

```
298 INPUT "PERCENTUALE SCONTO ";S
    C
299 RF$=""
300 PRINT
301 PRINT"RIFERIMENTI :"
302 INPUT RF$
303 WW$="CONTANTI"
304 PRINT
305 INPUT "TIPO DI PAGAMENTO ";WW
    $
306 PRINT
307 PRINT
308 INPUT "CONFERMA (0=SI * 1=NO)
       0[3 LEFT]";IC
309 IF IC<>0 THEN 279
310 RETURN
311 REM ***********************
312 REM * PREPARAZIONE COSTANTI*
314 REM ***********************
315 DIM RI$(27)
316 NC=3
317 CO$(1)="[DOWN]COPIA PER IL CL
    IENTE[UP]"
318 CO$(2)="[DOWN]COPIA PER LA CO
    NTABILITA'[UP]"
319 CO$(3)="[DOWN]COPIA PER USO I
    NTERNO[UP]"
320 CO$(4)="[DOWN]COPIA PER......
    ......[UP]"
321 R1$="":REM PREPARAZIONE FINC
    ATURA ORIZZONTALE
322 FOR I=1 TO 73
323 R1$=R1$+"-"
324 NEXT
325 BL$="":REM PREPARAZIONE STRI
    NGA A SPAZI
326 FOR I=1 TO 80
327 BL$=BL$+" "
328 NEXT
329 REM ***********************
330 REM *    COSTANTI PER LA    *
331 REM * STAMPANTE (MPS 801) *
332 REM ***********************
333 NN$=CHR$(15):REM DISABILITAZ
    IONE GRAFICI
334 MM$=CHR$(8):REM ABILITAZIONE
     GRAFICI
335 AA$=CHR$(14):REM ALLARGAMENT
    O CARATTERI
336 ZZ$="   [UP]D[DOWN]ESCRIZION
    E [UP]A[DOWN]RTICOLO
            [UP]P[DOWN]REZZO UN
    IT. "
337 ZZ$=ZZ$+"N.[UP]P[DOWN]EZ. [UP
    ]P[DOWN]REZZO [UP]T[DOWN]OT."
338 REM ***********************
339 REM * GENERALITA' FATTURA *
340 REM ***********************
341 INPUT "[CLEAR][3 DOWN][3 RIGH
    T]DATA FATTURA   0[3 LEFT]";
    F$
342 INPUT "[2 DOWN][3 RIGHT]NUMER
    O FATTURA   0[3 LEFT]";NF
343 INPUT "[2 DOWN][3 RIGHT]NUMER
    O COPIE  3[3 LEFT]";NC
344 INPUT "[4 DOWN]POSIZ. IL MODU
    LO E PREMERE RETURN   0[3 LEF
    T]";IC
345 RETURN
```

i-2-K H

S.p.a di Pippoli Pippo
Via Pipponio n.46 - 20121 Milano
Tel.(02) 99.99.999 - 66.66.666 telex n.444444
C.F.e P I.V.A. n.1234567890

Fattura 276

Spett.
GIULIO MARCOZZI
C. BARONI 148
20142 - MILANO (MI)

Milano 31 luglio 1984
Riferimenti Ns.bolla n.12345
Vostro C.F e/o P.Iva n. ABCDEFGHIJ

Descrizione Articolo	Prezzo unit.	n.Pez.	Prezzo Tot.
C 64	625.000	1	625.000
1520 PLOTTER	375.000	1	375.000
MPS 801 STAMPANTE	515.000	1	515.000
1530 REGISTRATORE	120.000	1	120.000
1541 FLOPPY DRIVE	630.000	1	630.000
1702 MONITOR A COLORI	690.000	1	690.000
1311 JOYSTICK	13.500	2	27.000
1312 PADDLE	22.500	1	22.500
D3 DISCHETTI 5"	5.000	20	100.000
C 20 CASSETTE	1.500	30	45.000
801 R NASTRO PER MPS 801	16.000	3	48.000
6068 SUPERBASE	175.000	1	175.000
6045 CALC RESULT	195.000	1	195.000
6014 DIARY 64	95.000	1	95.000
64207 EASY SCRIPT	75.000	1	75.000
64108 SIMONS' BASIC	85.000	1	85.000
6900 GUIDA AL COMMODORE 64	28.000	1	28.000
6804 PROGRAMMER'S REFERENCE GUIDE	35.000	1	35.000

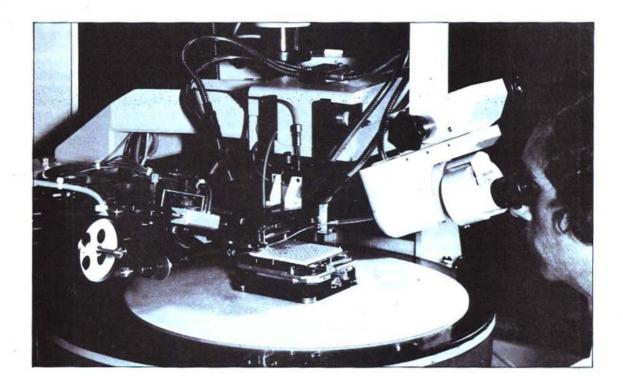
Tot.Imponib.	Sc. %	Tot. Scont.	IVA %	Tot. IVA	Tot Fatt.	
3.885.500	10	3.496.950	18	629.451		4.126.401

S.E.& O.

Modalita' di pagamento - rimessa diretta ricevimento fattura

copia per il cliente

TITOLATRICE
ovvero: come gestire le stringhe

di **Francesco Gatti**

Una stringa di caratteri, detta anche variabile alfanumerica, consiste in una qualunque sequenza di caratteri, di numeri o di simboli grafici, racchiusa fra virgolette. L'unico carattere grafico che richiede una particolare attenzione corrisponde alle doppie virgolette, infatti all'interno di una stringa non potrebbe comparire questo tipo di carattere perché viene normalmente usato per indicare l'inizio e la fine della stringa in oggetto.

Una semplice stringa è rappresentata da:

PRINT "FRANCESCO" (RETURN)

Nella stesura di un programma ci può tornare utile, come accade per i numeri, assegnare determinati messaggi a delle variabili permet-

tendo così di non dovere, ogni volta che ci occorrono, riscriverli per intero. La sintassi che permette questo è costituita da una lettera o due lettere qualunque dell'alfabeto seguita dal simbolo dollaro ($). È possibile usare nomi di variabili con più di due caratteri, ma questo accorgimento può servire esclusivamente per avere una maggior chiarezza nel nostro programma essendo solo le prime due lettere del nome-variabile riconosciute dal computer. È bene precisare che il secondo carattere può essere anche un numero, così é corretto definire: A$, AA$, A1$, A2$ etc. Ad esempio LIBRO$ e LI$ sono identici e individuano la stessa stringa. In ogni caso la variabile non può e non deve mai contenere parole chiave del BASIC.

Potremo avere infatti dei problemi con variabili tipo:

CORRENTISTA$.

In questo caso l'inteprete BASIC intenderebbe:

C OR RENTISTA$, e quindi una operazione di OR fra una variabile numerica (C) ed una alfanumerica (RENTISTA$) non è certo possibile.

Esempio di assegnazione:

```
10 A$="FRANCESCO"
20 PRINT A$
```

Una volta dato il comando RUN apparirà di nuovo la scritta che è stata posta tra le virgolette nella riga 10.

La possibilità di attribuire alla variabile A$ un certo commento alfanumerico ha molte applicazioni nel corso di un programma.

Tutte le istruzioni di INPUT possono essere effettuate con l'ausilio di STRINGHE, le quali possono fornire le informazioni di ciò che dovrà essere battuto. Se noi desideriamo che il contenuto della stringa A$ sia coerente possiamo affidarci ad un messaggio che invieremo all'operatore.

L'esempio che segue, in particolare alla riga 10, ci mostra un corretto uso di messaggio di IMPUT.

```
10 INPUT "COME TI CHIAMI";A$
20 IF A$="FRANCESCO" THEN 40
30PRINT "PIACERE": END
40 PRINT "BEL NOME"
```

Dopo il comando RUN il computer chiederà quale è il vostro nome, se voi risponderete con FRANCESCO salterà alla riga 40, in caso contrario eseguirà la 30; questo perchè ha riconosciuto nell astringa A$ i stessi caratteri prefissati nella riga 20.

Oltre a queste possibilità, il BASIC mette a sisposizione vari comandi per gestire una stringa: LEN (X$), LEFT$ (X$, X), RIGHT$ (X$, X), MID$(X$, X, Y), STR$ (X), VAL(X$), ASC(X$), CHR$(X)

LEN(X$) Se usato unitamente all'istruzione PRINT fornisce la lunghezza della stringa definita dal nome della variabile racchiusa fra parentesi. Questo comando può essere usato sia in modo diretto che in programma:

```
10 INPUT "COME TI CHIAMI"; A$
20 X=LEN (A$)
30 IF X)10 THEN 50
40 PRINT "PIACERE": END
50 PRINT "CHE NOME LUNGO HAI"
```

LEFT$(X$, X) Prende in considerazione solamente la parte sinistra di una stringa (X$) di X caratteri.

```
10 INPUT "COME TI CHIAMI";A$
20 PRINT LEFT$6A$, 1)
```

Questo semplice programma stamperà solo l'iniziale del vostro nome.

RIGHT$(X$, X) Come per il comando LEFT$(, con questo comando si otterrà l'effetto di prendere in considerazione la parte destra della stringa X$ di X caratteri. Sostituite questo comando con quello della riga 20 del precedente esempio e scoprirete cosa succede.

MID$(X$, X, Y) Prende in considerazione una parte della stringa in questione (X$) partendo dalla posizione X per Y caratteri (il parametro Y può essere omesso: in tal caso il valore di Y sarà equivalente al numero di caratteri che mancano al termine della stringa):

```
10 INPUT "COME TI CHIAMI":A$
20 Y=LEN(A$)—2: PRINT MID$(A$, 2, Y)
```

Come risultato avremo sullo schermo il nostro nome senza la prima e ultima lettera.

ASC(X$) Questa funzione fornisce il codice ASCII del primo carattere di X$.

CHR$(X) Esegue esattamente l'effetto opposto della precedente istruzione, fornisce cioè un carattere alfanumerico il cui codice numerico è X.

A cosa servono questi comandi?

Ora mostrerò delle semplici applicazioni di questi comandi che permettono di trattare sequenze di caratteri come più ci aggrada. Le stringhe nei computers COMMODORE ammettono, oltre a tutte le lettere e numeri, anche i simboli grafici; è così possibile inserire codici colori per stampare stringhe variopinte, ecc. Questo torna utile se si vuole far lampeggiare una scritta sul nostro schermo.

LISTATO K

Dato il comando RUN e inserito il nostro messaggio, questo comincerà a lampeggiare. Ciò è dovuto essenzialemente alla riga 30. Ogni volta che tale riga è in esecuzione i comandi RIGHT$ e LEFT$ i simboli di REVERSE/ON e REVERSE/OFF vengono scambiati continuamente fra loro ottenendo così ora un'accensione, ora uno spegnimento. Quindi la riga 40 accoppia la stringa R$ con A$.

Proviamo a far ruotare sullo schermo una scritta! Vediamo teoricamente cosa bisogna fare: presa una stringa dobbiamo stamparla partendo dal suo secondo carattere più il primo messo nell'ultima posizione, innescando un LOOP otterremo così lo scivolamento dei caratteri verso sinistra.

Il programma precedente è un po' limitato dalla lunghezza della stringa: infatti se noi introduciamo più di 39 caratteri ne succedono di tutti i colori! Allora con il programma seguente si può ovviare a questo problema, non solo, ma ci si può permettere di definire una finestra dentro la quale la nostra stringa scorrerà.

L'ampiezza della finestra può essere modificata cambiando il valore contenuto nella riga 10, introdurre quindi le coordinate (verticale e orrizzontale rispettivamente nelle righe 90 e 80) per definire la zona di partenza della stringa e per aumentare o diminuire lo 'scrolling' si deve premere una o più volte il tasto '+' o '—' a seconda dell'occorenza.

LISTATO K6

E se volessimo eseguire esattamente il contrario, cioè ottenere lo spostamento dei caratteri da sinistra verso destra, e non solo, ma quando lo spostamento arriva al margine destro si abbia una inversione dell'effetto allora dobbiamo accoppiare due movimenti in un soppio LOOP che, una volta spostata la nostra stringa sino a destra la riporta a sinistra e così via.

Otterremo un simpatico effetto che ci potrà tornar utile pe abbellire i nostri programmi

LISTATO K5

Ora se riuniamo le nostre conoscenze non dovrebbe essere difficile realizzare una titolatrice o, per intenderci meglio, un programma che fa "girare" sul nostro schermo un testo nello stesso modo con cui funzionano quelle scritte che vediamo per televisione che fanno scorrere i testi per essere comodamente letti (a titolo di curiosità questi titolatori/suggeritori si chiamano in gergo televisivo 'gobbi').

L'idea più funzionale è di trattare il nostro testo con un INPUT diretto usando un GET. Dovremmo perciò introdurre da tastiera, quale ultimo carattere della stringa, un FLAG, un carattere particolare che avverta che abbiamo terminato di inseguire il nostro testo.

Ho scelto volutamente, e non per comodità, il tasto funzionale 'F1' (CHR$(133)) perchè così facendo è possibile usare tutti i tasti della tastiera COMMODORE per poter anche creare dei piccoli disegni, nulla vieta però di utilizzare un qualsiasi altro carattere, modificando opportunamente il programmino.

Il programma inizia così con un GET nella riga 150 e fino a che non si batterà alcun tasto rimmarrà in LOOP« di attesa. Se si dovesse battere il tasto 'F1' (riga 160), inizierà l'OUTPUT del nostro testo.

La pulizia dello schermo nel nostro testo avviene in una maniera un pò diversa del solito, infatti si esegue per 25 volte l'istruzione PRINT, ciò ci permette di eseguire un corretto SCROOLING verso l'alto.

Ho introdotto inoltre nella riga 320 cinque spazi nello stesso modo per distanziare la serie continua del nostro testo che mano a mano scorrerà sempre verso l'alto.

La velocità di questo scorrimento è controllata dalla routine di riga 260-330: se batteremo '+' il nostro testo sarà stampato con maggior velocità, se premeremo '—' avverrà il contrario.

K

```
10 INPUT  "STRINGA";A$
20 R$="[RVS][RVOFF]"
30 R$=RIGHT$(R$,1)+LEFT$(R$,1)
40 PRINT"[CLEAR]"R$A$
50 FOR A=0 TO 120:NEXT:GOTO 30
```

K1

```
10 INPUT  "STRINGA";A$
20 B$=A$+"
   "
30 B$=MID$(B$,2)+LEFT$(B$,1)
40 PRINT"[CLEAR]"B$
50 FOR T=0 TO 120:NEXT:GOTO 30
```

K5

```
10 INPUT  "STRINGA";A$
20 FOR I=1 TO 40:BL$=BL$+" ":NEXT
30 B$=LEFT$(A$+BL$,40)
40 FOR R=1 TO 40-LEN(A$)
50 B$=RIGHT$(B$,1)+LEFT$(B$,39)
60 PRINT"[HOME]"B$
70 NEXT
80 FOR L=1 TO 40-LEN(A$)
90 B$=RIGHT$(B$,39)+LEFT$(B$,1)
100 PRINT"[HOME]"B$
110 NEXT
120 GOTO 40
```

K6

```
10 PRINT"[CLEAR]DIGITAZIONE STRING
   A (*=FINE)"
20 INPUT X$
30 IF LEN(XX$)+LEN(X$)>255 THEN 20
40 IF RIGHT$(X$,1)="*" THEN XX$=XX
   $+LEFT$(X$,LEN(X$)-1):GOTO 70
50 XX$=XX$+" "+X$
60 GOTO 20
70 PRINT"[CLEAR]"
80 HT=5:REM  HORIZZONTAL TAB
90 VT=5:REM  VERTICAL    TAB
100 LS=15:REM  LENGHT STRING
110 DL=105:REM  DELAY
120 VT$="[HOME][23 DOWN]"
130 PRINTLEFT$(VT$,VT) TAB(HT)LEFT$
    (XX$,LS)
140 XX$=MID$(XX$,2)+LEFT$(XX$,1)
150 GET N$
160 IF N$="-" THEN DL=DL+10
170 IF N$="+" THEN DL=DL-10
130 IF DL<1 THEN DL=1
190 FOR I=1 TO DL:NEXT:GOTO 130
```

KS

```
10 REM ******************************
   *******
20 REM ***
   ***
30 REM ***          TITOLATRICE
   ***
40 REM ***
   ***
50 REM ***         FRANCESCO GATTI
   ***
60 REM ***        MILANO 02/4691863
   ***
70 REM ***
   ***
80 REM ******************************
   *******
90 POKE 53280,2:POKE 53281,2:PRINT
   "[CLEAR]"
100 PRINT"[3 DOWN][RVS][GIALLO] UNA
    VOLTA TERMINATO IL VOSTRO TEST
   O [RVOFF]"
110 PRINT TAB(8)"[3 DOWN][RVS][BIAN
   CO] PREMERE 'F1' [RVOFF]"
120 PRINT"[4 DOWN] PER AUMENTARE O
   DIMINUIRE LA VELOCITA'"
130 PRINT"[2 DOWN] DELLA TITOLATRIC
   E PREMERE '+' O '-'"
140 FOR Y=0 TO 9000:NEXT:PRINT"[CLE
   AR]"
150 GET X$:IF X$="" THEN 150
160 IF X$=CHR$(133) THEN 210
170 PRINTX$;
180 XX$=XX$+X$
190 IF LEN(XX$)>200 THEN X$(N)=XX$:
   N=N+1:XX$=""
200 GOTO 150
210 X$(N)=XX$
220 FOR I=1 TO 25:PRINT:NEXT
230 FOR I=0 TO N
240 FOR L=1 TO LEN(X$(I))
250 REM ***   DELAY     ***
260 FOR D=1 TO DL:NEXT
270 GET S$:IF S$="+" THEN DL=DL-10
280 IF S$="-" THEN DL=DL+10
290 IF DL<1 THEN DL=10
300 PRINTMID$(X$(I),L,1);
310 NEXT:NEXT
320 FOR T=1 TO 5:PRINT:NEXT
330 GOTO 230
```

BIT MAP: EQUAZIONI E GRAFICA CON IL 64

di Giancarlo de Cobelli

La cosa più bella che può offrire un home computer è senz'altro il fatto di poter disegnare in alta risoluzione. Cioè poter definire ogni punto dello schermo per mezzo di comandi appropriati con diversi colori e come meglio si voglia.

Anche il Commodore 64 offre questa possibilità: utilizzando il modo Bit Map noi possiamo dare ad ogni punto (pixel) dello schermo un valore alla locazione di schermo corrispondente (bit). Ponendo a livello logico "uno" il bit in questione si accende il pixel scelto, ponendolo a zero lo si spegne. Quando noi passiamo in Bit Map visualizziamo sullo schermo una sezione di memoria di 8000 bytes, questo perché lo schermo è formato da 64.000 punti distinti indirizzabili da noi; infatti, sappiamo che un byte su un computer tipo il Commodore 64 è fatto da 8 bit, quindi 8.000x8 è proprio uguale a 64.000 bit.

Il Commodore 64 visualizza un carattere utilizzando una matrice di 8 x 8 bytes; sapendo che ci sono 25 linee e 40 colonne in alta risoluzione avremo disponibili 320 pixel orizzontali per 200 pixel verticali.

1) Modo Bit Nap standard (320x200 punti)
2) Modo Bit Nap multicolore (160x200 punti)

Possiamo subito notare che il modo standard propone una maggiore risoluzione orizzontale a discapito però della risoluzione a colori. Il modo Bit Map standard viene attivato impostando ad 1 il bit BMM del registro 17 ($11 Hex) dell'integrato 6566/6567, il famoso VIC II.

L'accesso alla matrice video viene modificata perchè ora i dati interpretati come puntatori di caratteri sono usati come dati colore. Quindi l'indirizzo per prelevare i dati dal byte 8000 della base video è organizzato nel seguente modo:

A13	A12	A11	A10	A09	A08	A07
A06	A05	A04	A03	A02	A01	A00
CB13	VC9	VC8	VC7	VC6	VC5	VC4
VC3	VC2	VC1	VC0	RC2	REC1	RC0

VCx è il contatore della matrice video; RCx è il contatore della linea di quadro; CB13 è il bit alto del puntatore di caratteri che si trova nel registro 24 (18 Hex) RCx, essendo formato da tre bit, fornisce la matrice 8 x 8. Difatti mentre si incrementa di uno per ogni linea orizzontale il contatore della matrice video punta le stesse 40 locazioni della memoria di schermo per otto linee (RCx) di quadro continuando poi sulle 40 successive locazioni per un totale di 320 punti ogni linea di quadro. Ad ogni incremento di RCx questo indirizzamento viene formattato sullo schermo con una matrice di 8 x 8 punti.

Per attivare la Bit Map Standard bisogna digitare la seguente POKE:
POKE 53265, PEEK (53265) OR 32
il che equivale ad impostare ad uno il bit 5 della locazione di memoria 53265.

Per disattivare la Bit Map Standard bisogna settare il bit 5 e ciò si fa digitando la POKE:
POKE 53265, PEEK (53265) AND 233

Come detto prima, le locazioni di schermo ora vengono utilizzate per controllare il colore del singolo punto. I colori delle matrici 8 x 8 che formano lo schermo non provengono dalla memoria colore bensì dalla memoria di schermo. Gli otto bit sono divisi in due semibyte per poter così usare due colori in ogni matrice 8 x 8; i quattro bit più alti della memoria di schermo diventano il colore di tutti i bit impostati ad uno ed i quattro bit più bassi della memoria di schermo di quelli impostati a zero.

Utilizzando lo schermo ad alta risoluzione, la Bit Map viene locata all'indirizzo 4096. Per avere a disposizione un'area sufficientemente grande di memoria RAM da utilizzare per scrivere i nostri programmi in linguaggio BASIC dovremo spostare la Bit Map in un'altra zona di memoria altrimenti ci resterebbero solo 2048 bytes (2 Kb) per il BASIC

Per poter fare questo bisogna utilizzare la locazione di memoria 53272 ($DO18 Hex) e porre ad uno il bit 3 (corrisponde a CB13) difatti questo registro che è interno al VIC III è il puntatore di memoria.

Con la POKE seguente possiamo utilizzare per i nostri programmi BASIC circa 6 Kb di RAM:

POKE 53272, PEEK (53272) OR 8

Se ci fosse bisogno di altro spazio di memoria occorre spostare la parte bassa del BASIC oltre l'area di memoria occupata dallo schermo.

Naturalmente tutti questi problemi non si verificano programmando in linguaggio macchina, poiché ci sono altre zone di memoria utilizzabili. Difatti la maggior parte di operazioni in alta risoluzione sono dei procedimenti ripetitivi che rendono il BASIC molto lento e si prestano invece molto bene per l'uso del linguaggio macchina. La soluzione migliore è quella di utilizzare il linguaggio macchina per le procedure in alta risoluzione richiamandole poi da BASIC con una SYS XXXXX.

La "bit mapping" è organizzata su un sistema di assi cartesiani con l'origine nel punto in alto a sinistra dello schermo (locazione 1024). Quindi il massimo valore della X sarà nell'angolo in alto destra e varrà 319; quello della Y sarà nell'angolo in basso a sinistra e varrà 199.

La pagina grafica (essendo organizzata in 25 righe di 40 colonne ciascuna) presenta delle difficoltà nel calcolo dell'accensione o spegnimento di un punto poiché la numerazione dei pixel non è sequenziale né in orizzontale né in verticale. Per esempio la colonna 1 riga o è così organizzata:

byte 0 byte 8 byte 16 ... byte 312
byte 1 byte 9 byte 17 ... byte 313
byte 2 byte 10 byte 18 ... byte 314
byte 3 byte 11 byte 19 ... byte 315
byte 7 byte 15 byte 23 ... byte 319

Comunque con l'utilizzo di una formula, il calcolo viene semplificato. La formula consiste nel fissare la base (inizio area di memoria schermo), calcolare la riga, la colonna, la linea (byte) della posizione in cui si trova il carattere e sommarle. Ponendo X e Y le coordinate degli assi abbiamo:

Riga = INT (Y/8)
Colonna = INT (X/8)
Linea = Y AND 7
Byte = Base + Riga *320 + Colonna *8 + Linea

Quale bit sarà accesso di quel Byte? Il risultato è dato dalla formula:

Bit = 7 — (X AND 7)

Ora per sapere quale Bit della matrice di coordinate X, Y sarà attivato basterà digitare la formula:

POKE Byte, PEEK (Byte) OR (2 Bit)

Prima di usare lo schermo in alta risoluzione bisogna pulire la sezione di memoria che abbiamo interessato ed impostare i colori che intendiamo utilizzare.

Con l'istruzione seguente puliamo la pagina grafica della Bit map:

FOR E = base TO Base + 7999: POKE E, 0: NEXT

La scelta dei colori è possibile tramite i due semibytes in questo modo:

- il semibyte alto, corrispondente ai quattro bit più significativi, serve per definire il colore dei punti
- il semibyte basso, i quattro bit meno significativi servono per definire il colore del fondo.

La formula esemplificativa da inserire nella POKE è data dalla moltiplicazione del codice colore dei punti per 16 più il codice colore del fondo. Ad esempio per settare punti verdi su sfondo nero basterà digitare:

FOR C = 1024 TO 2023: POKE C, 80: NEXT

Una tipica applicazione dello schermo in alta risoluzione nel modo standard è la visualizzazione di funzioni matematiche.

Il listato del programma proposto permette la visualizzazione di queste funzioni sugli assi cartesiani e quindi uno sviluppo su quattro quadranti oppure solamente sul primo.

Il programma permette di visualizzare tutte le funzioni che gli verranno proposte definendo il campo di esistenza (estremo inferiore ed estremo superiore), la scala degli assi cartesiani X, Y il passo (più piccolo è e più sarà precisa la rappresentazione grafica anche se diminuirà la velocità di tracciamento); per default sono stati inseriti dei valori che sono ritenuti i più opportuni per una prima visualizzazione della funzione.

Bisogna ricordarsi di non inserire valori negativi per l'estremo inferiore nel caso si scegliesse la rappresentazione solo sul primo quadrante e che alla fine di ogni rappresentazione bisogna digitare RUN STOP / RESTORE. Il programma comunque si presta a molte modifiche.

Ad esempio, si può inserire una routine di hardcopy dello schermo. Bisogna solo stare attenti di non superare come memoria BASIC la locazione 8192 altrimenti pulendo la pagina in alta risoluzione verrebbe cancellata anche quella parte del programma.

Descrizione del programma

1 Definizione del colore e passaggio al modo Minuscolo/Maiuscolo con il codice ASCII 14; il codice ASCII 18 disabilita i tasti SHIFT ed il tasto con il logo Commodore.

1-2-26 Intestazione del programma e accettazione dei valori di input; dalla riga 21 c'è la routine di scelta tra il primo od i quattro quadranti.

32-36 La subroutine di inizializzazione viene utilizzata per abilitare la Bit Map, locarla all'indirizzo di memoria 8192, per ripulire la pagina

grafica e per definire il colore dei punti e del fondo.

42-44 Subroutine di calcolo delle coordinate del punto da accendere.

50-68 le righe 51 e 52 tracciano rispettivamente l'asse X e l'asse Y; le righe 53-56 quotano gli assi; le righe 57-60 scrivono "X" e "Y" sul video in corrispondenza degli assi; le righe 61-67 provvedono a modificare i parametri ed a calcolare il valore della funzione punto per punto.

74-88 Stessa funzione delle righe sovraesposte con i parametri riferiti al primo quadrante.

89 Segnala con la modifica del colore dell'angolo in alto a sinistra la fine dell'elaboratore della funzione.

90-93 Aspetta la battitura di un tasto per disabilitare la Bit Map e concludere così il programma.

1-94 Linea contenente la funzione da analizzare e che naturalmente può essere modificata a piacimento rispettando la corretta sintassi richiesta dal Commodore 64 per quanto riguarda le espressioni matematiche.

Per quanto riguarda invece il modo Bit Map Multicolore oltre alle due POKE viste precedentemente, ovvero la POKE53265 e la POKE53272, occorre porre ad uno il bit MCM del registro 22 ($16 HEx).

La seguente POKE provvede a ciò:

POKE 53270, PEEK (53270) OR 16

Per disattivarla basterà settare il bit impostato precedentemente con il valore uno con la POKE seguente:

POKE 53270, PEEK (53270) and 239

Questo modo di procedere permette l'utilizzo di quattro colori per ogni matrice da 8 bit. Naturalmente la risoluzione orizzontale viene dimezzata poiché ora servono due bit per rappresentare un punto sullo schermo. Avremo in questo modo 160 punti orizzontali per 200 punti verticali.

```
1 POKE 53280,12:POKE 53281,12:PRI
  NTCHR$(14):PRINTCHR$(8)
2 PRINT"[CLEAR][4 DOWN][BLEU]
  "
3 PRINT"[2 DOWN][GIALLO]
    ♠TUDIO DI ⌊UNZIONI"
4 PRINT"[DOWN]                DI ⌐E ⌐OBELL
  I |IANCARLO <->"
5 PRINT"[DOWN][ROSSO]                T
  EL. 02/6192306"
6 PRINT"[DOWN][BLEU]
  "
7 FOR R=1 TO 1800:NEXT
8 GOTO 9
9 PRINT"[CLEAR]"
10 PRINT"[HOME]":PRINT "[RVS]
    ⌐NSERIRE I PARAMETRI RICHIESTI
      [RVOFF]"
11 PRINT"[3 DOWN]♠CALA ♠
    = 30[4 LEFT]";:INPUT SX:IF SX<
   =0 THEN 10
12 PRINT"[DOWN]♠CALA |                =
    70[4 LEFT]";:INPUT SY:IF SY<=0
    THEN 10
13 PRINT"[DOWN]⌐ASSO                  =
    .5[4 LEFT]";:INPUT PA:IF PA<=0
    THEN 10
14 PRINT"[DOWN]⌐STREMO ⌐NFERIORE =
    -5[4 LEFT]";:INPUT EI
15 PRINT"⌐STREMO ♠UPERIORE =  5[4
   LEFT]";:INPUT ES
16 PRINT"[DOWN]⌐OLORE ⌊UNZIONE     =
    5[4 LEFT]";:INPUT FU:IF FU<0
   OR FU>15 THEN 10
17 PRINT"⌐OLORE ⌊ONDO          =  0[4
   LEFT]";:INPUT FO:IF FO<0 OR FO>
   15 THEN 10
18 PRINT"[DOWN]⌐ PARAMETRI INSERIT
   I SONO GIUSTI? (♠/⁄)"
19 GET SN$:IF SN$="" THEN 19
20 IF SN$="N" THEN 10
21 PRINT"[3 DOWN]⌐ER STAMPA SU PRI
   MO   QUADRANTE BATTI [RVS]1[RVO
   FF]"
22 PRINT"[DOWN]⌐ER STAMPA SU QUATT
   RO QUADRANTE BATTI [RVS]4[RVOFF
   ]"
23 GET Q$:IF Q$="" THEN 23
24 IF VAL(Q$)=1 THEN 74
25 IF VAL(Q$)=4 THEN 50
26 IF VAL(Q$)<>1 OR VAL(Q$)<>4 THE
   N 23
27 REM ****************************
   *****
28 REM ***
     ***
29 REM *** SUB. DI INIZIALIZZAZION
   E ***
30 REM ***
     ***
31 REM ****************************
   *****
32 PRINT"[CLEAR]":POKE 53272,PEEK(
   53272) OR 8
33 POKE 53265,PEEK(53265) OR 32
```

La selezione dei quattro colori avviene attraverso il registro o del colore di fondo, dalla memoria colore, dalla matrice video (i 4 bit più alti definiscono un colore, i 4 più bassi un altro).

L'accesso alla memoria video avviene come nel modo Standard, come del resto tutte le altre procedure, solo che il punto specificato verrà interpretato nel seguente modo:

la coppia di bit 00 seleziona il colore di fondo preso dal registro 33;

la coppia di bit 01 seleziona il colore del punto preso dal semybite alto della matrice video;

la coppia di bit 10 seleziona il colore del punto preso dal semibyte basso della matrice video;

la coppia di bit 11 seleziona il colore del punto preso dal semibyte colore della matrice video.

A questo punto potete avventurarvi nel bellissimo mondo dell'alta risoluzione creando, per le vostre realizzazioni, effetti più funzionali e... più fantasiosi.

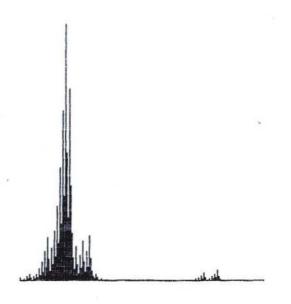
```
34 FOR P=8192 TO 16191:POKE P,0:NE
   XT
35 FOR C=1024 TO 2023:POKE C,(FU*1
   6+FO):NEXT
36 RETURN
37 REM ****************************
39 REM ***       SUB. DI CALCOLO
41 REM ****************************
42 CL=INT(X/8):RG=INT(Y/8):LN=Y AN
   D 7:BY= 8192+320*RG+8*CL+LN
43 BI=7-(X AND 7)
44 POKE BY,PEEK(BY) OR (2↑BI):RETU
   RN
45 REM ****************************
47 REM ***     QUATTRO QUADRANTI
49 REM ****************************
50 GOSUB 32
51 FOR X=5 TO 314:Y=100:GOSUB 42:N
   EXT
52 FOR Y=5 TO 195:X=160:GOSUB 42:N
   EXT
53 FOR X=160 TO 314 STEP SX:Y=103:
   GOSUB 42:NEXT
54 FOR X=160 TO 5 STEP (-SX):Y=103
   :GOSUB 42:NEXT
55 FOR Y=100 TO 5 STEP (-SY):X=157
   :GOSUB 42:NEXT
56 FOR Y=100 TO 195 STEP SY:X=157:
   GOSUB 42:NEXT
57 FOR I=0 TO 9:READ X,Y:GOSUB 42:
   NEXT
58 DATA 309,103,313,103,310,104,31
   2,104,311,105,310,106,312,106,3
   09,107,313,107
59 FOR I=0 TO 5:READ X,Y:GOSUB 42:
   NEXT
60 DATA 152,6,156,6,153,7,155,7,15
   4,8,154,9,154,10
61 EI=EI*SX+160:ES=ES*SX+160
62 IF EI<1 THEN EI=1
63 IF ES>319 THEN ES=319
64 FOR X=EI TO ES STEP PA:X1=(X-16
   0)/SX:GOSUB 94:Y=INT(100-SY*(Y1
   ))
65 IF Y<=0 OR Y>199 THEN 67
66 GOSUB 42
67 NEXT
68 GOTO 89
69 REM ****************************
71 REM ***         PRIMO QUADRANTE
73 REM ****************************
74 GOSUB 32
75 FOR X=5 TO 314:Y=190:GOSUB 42:N
   EXT
76 FOR Y=5 TO 195:X=10:GOSUB 42:NE
   XT
77 FOR X=10 TO 314 STEP SX:Y=193:G
   OSUB 42:NEXT
78 FOR Y=190 TO 5 STEP (-SY):X=7:G
   OSUB 42:NEXT
79 FOR I=0 TO 9:READ X,Y:GOSUB 42:
   NEXT
80 DATA 309,193,313,193,310,194,31
   2,194,311,195,310,196,312,196,3
   09,197,313,197
81 FOR I=0 TO 5:READ X,Y:GOSUB 42:
   NEXT
82 DATA 3,6,7,6,4,7,6,7,5,8,5,9,5,
   10
83 EI=EI*SX+5:ES=ES*SX+5:Y=193
84 IF ES>309 THEN ES=309
85 FOR X=EI TO ES STEP PA:X1=(X-10
   )/SX:GOSUB 94:Y=INT(190-SY*(Y1)
   )
86 IF X<10 OR Y>190 THEN 88
87 GOSUB 42
88 NEXT
89 POKE 1024,(FU*16+(FO+1))
90 GET R$:IF R$="" THEN 90
91 POKE 53265,PEEK(53265) AND 233
92 POKE 53272,PEEK(53272) AND 247
93 END
94 Y1=SIN(X1):RETURN
```

CHICCHI DI GRANDINE: UN CURIOSO ED IRRISOLTO PROBLEMA DELLA TEORIA DEI NUMERI

di Mariangela Guardione e Eugenio Coppari

I chicchi di grandine, prima di cadere a terra, subiscono una serie di fenomeni fisici dovuti alla temperatura, alla massa, alla gravità e alla forza ascensionale delle correnti d'aria.

La grandine nasce ad una determinata altitudine per effetto termico e quindi tende a cadere per gravità; a questo punto se la massa dei chicchi non è tale da contrastare le correnti ascensionali gli stessi tendono a risalire acquistando nel contempo massa aggiuntiva sino a che il loro peso non è nuovamente tale da non subire più l'effetto della forza ascensionale. Ricadendo si può ripresentare ancora il fenomeno precedente e così via.

Nel numero di marzo 1984 della rivista le SCIENZE è apparso nella

parte relativa alle rubriche, in particolare quella inerente all'informatica, un articolo molto interessante di BRIAN HAYES nel quale viene trattato un curioso ed irrisolto problema della teoria dei numeri che può essere così spiegato:

si scelga un numero intero positivo qualsiasi e lo si chiami N.

Se il numero N. è pari lo si deve dividere per 2 (N/2).

Se il numero N. è dispari lo si moltiplica per 3 e si somma 1 (3*N + 1).

In entrambi i casi, il risultato è il nuovo valore del numero N e si ripete la procedura fintantoché N non sia uguale a 1.

A questo punto le domande che possono sorgere sono:

— i numeri tendono a crescere o a decrescere?

— convergono su qualche valore particolare oppure divergono all'infinito?

Se si fa qualche esempio numerico prendendo un qualsiasi N (esempio N = 31), numero dispari, lo si deve moltiplicare per 3, sommargli 1 e si ottiene quindi il valore successivo 94, che è pari e va perciò diviso per 2 con risultato 47, che è dispari; per cui si ricomincia la procedura.

Ovviamente ci saranno molti alti e bassi in questa serie di numeri: il valore cresce quando N è dispari mentre decresce quando è pari. Il difficile non sta nel calcolare la serie per un dato N, ma, nonostante moltissimi numeri siano stati sottoposti a verifica, nessuno è ancora riuscito a trovare una soluzione generale valida per tutti i possibili valori numerici di N.

Nonostante queste difficoltà il problema fornisce un buon esempio di come il calcolatore sia utile come strumento di calcolo matematico, in quanto rappresenta l'unico mezzo per trattare, in un tempo relativamente breve, numeri superiori ai più piccoli interi.

N	LUNGHEZZA DEL PERCORSO	VALORE MASSIMO
1	0	1
2	1	2
3	7	16
6	8	16
7	16	52
9	19	52
18	20	52
25	23	88
27	111	9232
54	112	9232
73	115	9232
97	118	9232
129	121	9232
171	124	9232
231	127	9232
313	130	9232
327	143	9232
649	144	9232
703	170	250 504
871	178	190 996
1161	181	190 996
2223	182	250 504
2463	208	250 504
2919	216	250 504
3711	237	481 624
6171	261	975 400
10 971	267	975 400
13 255	275	497 176
17 647	278	11 003 416
23 529	281	11 003 416
26 623	307	106 358 020
34 239	310	18 976 192
35 655	323	41 163 712
52 527	339	106 358 020
77 031	350	21 933 016

Sequenza dei percorsi più lunghi fino a N = 100 000

Utilizzando il calcolatore in questo "fastidioso problema di calcolo", come viene definito dai matematici, ci si può domandare quale risultato si possa ottenere quando venga applicata ripetutamente la regola di trasformazione ad un numero qualsiasi.

A questo proposito sono state formulate tre ipotesi che si basano su un'analisi probabilistica del problema, mentre la serie dei numeri generati non è casuale.

La prima di queste ipotesi inizia dall'osservazione che se gli interi positivi pari e dispari sono in numero uguale, una qualsiasi serie di calcoli di un N qualsiasi dovrebbe assumere con la stessa frequenza sia valori dispari che pari.

La seconda ipotesi si basa sul fatto che tutte le volte che il calcolo giunge ad una potenza di 2 precipita immediatamente al valore 1. Quindi nel corso del calcolo si possono raggiungere valori molto elevati di N, ma ad un certo punto si deve verificare una rapida discesa.

Infine, la terza ipotesi è analoga alla seconda, ma porta ad una diversa conclusione. Si può osservare infatti che, ogni volta che il calcolo giunge ad un numero dispari, prosegue con una serie di numeri pari e rientra quindi in un percorso già compiuto. Tutto questo vuol dire che, siccome la scelta del passo successivo è del tutto deterministico, un valore che si ripresenta deve portare ad un ciclo ripetuto indefinitivamente.

Se si svolge il calcolo scegliendo casualmente dei numeri (esercizio che può essere facilmente eseguito dal lettore) si possono distinguere due grandezze interessanti che sono:

— il valore più alto o "PICCO" che N raggiunge durante il calcolo

— la lunghezza del percorso che rappresenta il numero totale di passaggi necessari per raggiungere il valore 1.

Se si rappresentano graficamente i numeri che si ottengono partendo da un certo valore iniziale, si può osservare che essi si dispongono come la traiettoria seguita da un chicco di grandine sotto l'azione delle correnti ascensionali nel corso di una tempesta per poi ricadere sotto l'effetto del proprio peso.

È questo il motivo per cui il problema della teoria dei numeri di cui abbiamo parlato in questo articolo è noto come il problema dei "NUMERI A CHICCO DI GRANDINE".

Presentiamo a conclusione di questo articolo una versione del programma relativo al problema sopra citato in linguaggio BASIC, stilata per il COMMODORE 64.

Il programma è strutturato in due

blocchi:

— il primo è relativo alla fase di digitazione del numero, al calcolo del valore raggiunto ad ogni passo e al numero totale di passi per precipitare al valore 1.

— il secondo blocco riguarda la rappresentazione grafica dell'andamento del numero preso in considerazione a mezzo di istogrammi a canne d'organo.

In questa fase si stabiliscono i due assi di riferimento: l'asse Y sul quale viene riportato il valore massimo raggiunto dal numero durante il suo percorso, mentre sull'asse X si trova il numero dei passi che il numero esegue prima di precipitare al valore 1.

A questo punto si considera il valore (VAL) raggiunto nel primo passo dal numero considerato (memorizzato nella prima posizione del vettore) e lo si visualizza sullo schermo alle coordinate 1, VAL.

Si ripete l'operazione per tutti i valori del vettore tracciando per ognuno la parte d'istrogramma che gli compete.

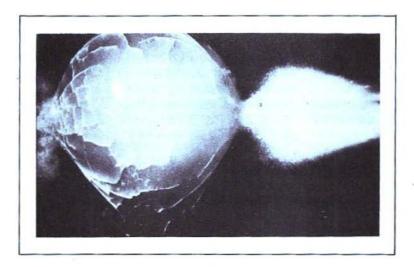
Si otterrà quindi l'andamento relativo al numero considerato e quindi l'effetto dei numeri che rimbalzano come chicchi di grandine.

```
CHICCHI DI GRANDINE


100 REM ************************
119 REM *    IL PROBLEMA DEI    *
120 REM *       CHICCHI DI      *
130 REM *        GRANDINE       *
140 REM ************************
150 REM *          DI           *
160 REM * MARIANGELA GLARDIONE *
170 REM *          E            *
180 REM *  EUGENIO COPPARI      *
190 REM ************************
200 DIM TT(320)
210 PRINT"[CLEAR][10 DOWN]"; TAB(
    15)"IL PROBLEMA"
220 PRINT"[DOWN]"; TAB(15)"DEI CH
    ICCHI"
230 PRINT"[DOWN]"; TAB(15)"DI GRA
    NDINE"
240 FOR TT=1 TO 4000:NEXT
250 INPUT "[CLEAR][4 DOWN]DIGITA
    UN NUMERO POSITIVO ";N
260 IF (N/2-INT(N/2))<.2 THEN  N=
    N/2:GOTO 280
270 N=3*N+1
280 F=F+1:TT(F)=N:PRINTTT(F):IF T
    T(F)>AA THEN 520
290 IF N>1 THEN 260
300 GOTO 310
310 FOR K=49152 TO 49222:READ P:P
    OKE K,P:NEXT
320 BA=8192:SYS49152
330 IF F>320 THEN END
340 IF F>160 THEN 530
350 UU=INT(320-2*F)/2
360 FOR GG=1 TO F
370 M=AA/TT(GG)
380 T=T+1
390 FOR ZZ=T TO T+1
400 FOR KK=199 TO 199-INT(199/M)+
    1 STEP -1
410 FF=INT((GG+UU+ZZ)/8)
420 JJ=INT(KK/8)
430 LN=KK AND 7
440 BY=BA+JJ*320+FF*8+LN
450 BI=7-((GG+UU+ZZ) AND 7)
460 POKE BY,PEEK(BY) OR (2↑BI)
470 NEXTKK
480 IF E=1 THEN RETURN
490 NEXTZZ
500 NEXTGG
510 GOTO 510
520 AA=TT(F):GOTO 290
530 E=1:UU=INT(320-F)/2
540 FOR GG=1 TO F
550 M=AA/TT(GG)
560 GOSUB  400
570 NEXT GG
580 GOTO 580
590 DATA  173,24,208,9,8,141,24,2
    08
600 DATA  173,17,208,9,32,141,17,
    208
610 DATA  169,0,133,251,169,32,13
    3,252
620 DATA  160,0,169,0,145,251,200
    ,192
630 DATA  0,208,249,230,252,169,6
    4,197
640 DATA  252,208,239,169,0,133,2
    51,169
650 DATA  4,133,252,160,0,169,3,1
    45
660 DATA  251,200,192,0,208,249,2
    30,252
670 DATA  169,8,197,252,208,239,9
    6
```

CHARACTER EDITOR

di Francesco Gatti

Questo programma vi consente di inserire a piacimento nuovi caratteri associati alla tastiera del vostro 64.

I caratteri del Commodore 64 sono formati da una matrice di 8 X 8 pixel.

Ma che cosa è un "pixel."

Avvicinatevi allo schermo quel tanto che vi permetta di esaminare bene, ad ésempio, il punto della "i" minuscola. In realtà, questo è costituito da due puntini chiamati "pixel". A seconda che essi siano accesi o spenti i pixel, determinano una "figura" ". In questo caso andranno a formare delle lettere, cifre, segni di punteggiatura ... ecc.

Torniamo al concetto della matrice 8x8. Abbiamo quindi otto file verticali composte da otto caselle (pixel). Ogni fila di PIXEL sarà un BYTE formato da 8 BIT, ciascuno dei quali può assumere esclusivamente i valori 0 o 1: assegnando il numero 0 il PIXEL sarà spento, mentre se assumerà 1 sarà acceso. Un carattere quindi sarà definito da 8 BYTE.

Per la programmazione della grafica bisogna conoscere innanzitutto il punto esatto da cui il VIC II (Video Interface Chip, ovvero Circuito Interfaccia Video, da non confondere con il fratello minore del Commodore 64, il Vic 20) prende le informazioni dei caratteri.

Normalmente questo circuito integrato preleva le caratteristiche di ogni carattere della ROM GENERATORE DI CARATTERI (ROM, come sapete è l'acronimo di Read Only Memory, ovvero memoria a sola lettura).

In questo circuito sono memorizzati tutti i dati che compongono i caratteri standard e, ogni

volta che si accende il computer, il Commodore esegue una particolare routine di inizializzazione (booth) che prepara, anche con i dati contenuti in ROM, la serie di matrice dei caratteri, esegue la prima "videata" di presentazione della macchina, abilita le operazioni inerenti alla gestione del output video. E così via. Una delle caratteristiche più importanti del Commodore 64 è la capacità di usare le configurazioni contenute nella RAM (= Random Access Memory ovvero memoria ed accesso causale), la cosiddetta memoria utente.

Queste configurazioni RAM possono perciò essere definite dall'utente ottenendo così un insieme quasi indefinito di simboli per giochi, applicazioni scientifiche, lingue straniere ecc.

É questa una cosa molto importante, perchè non usufruendo il 64 della possibilità di scelta del tipo di tastiera, ad esempio fra una QWERTY (tastiera americane) e una italiana (per QWERTY intendiamo la tastiera che ha sulla seconda riga le lettere q, w, e, r, t, y, mentre quella italiana detta Q2ERTY, ha la "z" al posto della "w", e la "m" a seguito della "l") con questo semplice programma possiamo, ogni volta che occorre, definire in RAM i caratteri desiderati e quindi richiamarli.

Nella ROM la memoria carattere inizia dalla locazione 53248. I primi 8 BYTEs, dalla locazione 53248 alla 53255, contengono la configurazione del simbolo (il cui codice ASCII è CHR$ (0)), gli 8 BYTEs successivi definiscono la lettera "a", dalla locazione 53256 alla 53263, e così via per tutti gli altri caratteri.

Il programma proposto fa esattamente ciò che è stato detto. Preleva cioè dalla ROM i caratteri standard, li visualizza e mediante una routine, permette di modificarli e memorizzarli in RAM.

REMarks

130 - Queste due POKEs permettono di definire una il colore del bordo dello schermo, l'altra il colore dello sfondo. Nel nostro caso avremo uno schermo completamente rosso (se volete cambiateli a piacere).
150 - Pone a zero la cella di memoria di lavoro
170 - 12288 è la locazione di memoria di partenza scelta per inserire i nuovi caratteri.
175 - La prima POKE esculde le interruzioni: se ad esempio si battesse accindentalmente qualche tasto la macchina reagirebbe in maniera anomala. La seconda POKE rende attivo un carattere vuoto.
190 - Determina il carattere e lo pone in maniera maiuscola (vedi tabella 1 per determinare altri tipi di scrittura)
205 - La prima POKE ripristina I/0. La seconda riattiva la tastiera.
210-220 - Viene "printata" sullo schermo una nuova riga, la 5, seguita dalla istruzione RUN. Nulla accade fino a quando non vengono eseguite le righe seguenti.
230 - 260 - La prima POKE determina il numero di caratteri del BUFFER della tastiera (in questo caso 3), le altre tre POKEs stabiliscono i risettivi caratteri da porre nel BUFFER: un ritorno in "HOME" (chr = 19) del cursore e 2 RETURN (chr=13). In questo modo è come se noi avessimo battuto una riga di programma seguita da un RETURN di conferma e quindi l'istruzione di RUN seguita sempre dal tasto RETURN, tutto però da programma.
280 - 480 - Nella riga 330 la funzione serve per visualizzare la locazione di POKE di carattere. Nella 340 la funzione definisce i valori della POKE di un carattere. Nella 350 si rimanda all'elenco di caratteri che si vogliono trattare, quindi dalla riga 360 si traccia il reticolato per la scelta dei caratteri.
510-610 - Questo blocco di istruizione si occupa di leggere il movimento del cursore per la scelta dei caratteri. Ciò avviene prendendo in esame il valore che assume la stringa A$ personalizzata dall'istruzione BASIC GET impostata nella riga 470.
710 - 700 - Riconoscimento dell'opzione scelta per la formazione di un nuovo carattere.
810 - 840 - Riconoscimento dell'opzione relativa alla seconda parte di operatività (schermo con griglia di creazione carattere).
1210 - 1300 - Visualizzazione delle opzioni per l'edit di un carattere
1520 - Questa POKE si occupa

della cancellazione dell'INTERO programma, rispristinando i caratteri standard.

1610 - 1700 - Istruzioni per la visualizzazione del valore di ogni singolo byte del carattere.

1810 - 1910 - Queste istruzioni "dividono" il carattere scelto in PIXEL

2010 - 2160 - Istruzioni che scrivono in coda al programma i DATA di ogni singolo carattere da noi definito. Potremo, in un secondo momento, rileggere sul il listato dalla linea 30000 in poi i valori relativi ai nuovi caratteri da noi generati e che potremo inserire ed utilizzare in altri programmi.

Come usare il programma

Dopo aver digitato il listato, "lanciatelo": sullo schermo apparirà una scritta che vi informerà che il computer sta prendendo le informazioni necessarie per la costruzione "punto a punto" dei caratteri. Pochi istanti più tardi sullo schermo verranno visualizzati la griglia dei caratteri e le opzioni del Character Editor:

a) formazione di un nuovo carattere al posto di uno vecchio;
b) modifica di un carattere già esistente;
c) soppressione del programma.

Selezionado la prima opzione, sulla griglia, dove erano disegnati i caratteri, scompariranno tutti i punti per lasciare posto ad un'altra griglia completamente vuota da puntini (questo spazio vuoto non sarà altro che il "buco" del carattere selezionato dalla griglia di partenza): è sopra questi punti che si dovrà definire il nostro nuovo carattere.

Completata la composizione si dovrà premere il tasto il simbolo (=), apparirà così a fianco di ogni BYTE i loro valori, quindi premere il tasto (D) per creare alla coda del programma linee di dati (DATA) caratteristici di ogni carattere inserito. Ora, ogni volta che si rilascerà il programma, il Commodore 64 utilizzerà i caratteri personalizzati.

Se avessimo premuto il tasto (E), quello cioè per l'opzione di EDIT, sarebbe stata visualizzata una griglia formata dai punti costituenti il carattere scelto con il cursore di selezione (CRSR), permettendo così di modificarlo "pixel to pixel".

Tabella 1

Come modificare il programma

Se si desidera modificare i caratteri non in modo maiuscolo basta cambiare la locazione di memoria della POKE contenuta nella riga 190 secondo i seguenti valori:

53248	Caratteri Maiuscoli
53760	Caratteri Grafici
54272	Caratteri Maiuscoli "reverse"
54784	Caratteri Grafici "reverse"
55296	Caratteri Minuscoli
55808	Caratteri Maiuscoli e Grafici
56320	Caratteri Minuscoli "reverse"
56832	Caratteri Maiuscoli e Grafici "reverse"

```
10 REM *************************      *****
   ********
11 REM ****   CHARACTER EDITOR        *****
12 REM ****   APRILE 1984
```

```
 13 REM ****        BY ITTAG
     *****
 14 REM ************************
    ********
130 POKE  53280,2 : POKE  53281,2
140 PRINT "[CLEAR][GIALLO]     *C
    OSTRUZIONE CARATTERI*
150 POKE  828,0
160  RUN 170
170 CS=12288
175 POKE  56334, PEEK (56334)  AN
    D  254 : POKE  1, PEEK (1)  A
    ND  251
180 FOR  I=CS  TO  CS+2047
190 POKE  I, PEEK (53248+I-CS)
200 NEXT  I
205 POKE  1, PEEK (1)  OR  4 : PO
    KE  56334, PEEK (56334)  OR
    1
210 PRINT  "[CLEAR] 5 RUN 280"
220 PRINT  "RUN"
230 POKE  198,3
240 POKE  631,19
250 POKE  632,13
260 POKE  633,13
270 END
280 S=1024 : CL=40
290 CS=12288
300 CR=0 : LN=30000+ PEEK (828)
310 P=24 : BG=1 : BR=1
320 POKE  53280,2 : POKE  53281,2
330 DEF   FNA (XX) =S+R*2*CL+2*C
340 DEF   FNB (XX)=8*R+C
350 GOTO  1000
360 PRINT  "[CLEAR][BIANCO]" : GO
    SUB  1200
370 PRINT  "[HOME]" ;: FOR  I=0
    TO  7
380 PRINT ". . . . . . . . " : PR
    INT
390 NEXT  : F=0
400 PRINT  "[HOME]" : R=0 : C=0
410 Z= FNA (0)
420 IF  F=0  THEN  460
430 IF  Z=ZL  THEN  450
440 POKE  ZL,IL : ZL=Z : IL= PEEK
     (ZL)
450 POKE  Z+54272,0
460 POKE  Z+54272,0
470 GET  A$ : IF  A$=""  THEN  47
    0
480 POKE  Z+54272,1
510 IF  A$= "Q"  THEN  1500
520 IF  A$= "[RIGHT]"  AND  C=7
    THEN  C=0 : GOTO  410
530 IF  A$= "[RIGHT]"  THEN  C=C+
    1 : GOTO  410
540 IF  A$= "[LEFT]"  AND  C=0  T
    HEN  C=7 : GOTO  410
550 IF  A$= "[LEFT]"  THEN  C=C-1
     : GOTO  410
560 IF  A$= "[DOWN]"  AND  R=7  T
    HEN  R=0 : GOTO  410
570 IF  A$= "[DOWN]"  THEN  R=R+1
     : GOTO  410
580 IF  A$= "[UP]"  AND  R=0  THE
    N  R=7 : GOTO  410
590 IF  A$= "[UP]"  THEN  R=R-1 :
     GOTO  410
600 IF  A$= "[HOME]"  THEN  400
610 IF  F=1  THEN  800
710 IF  A$= "+"  THEN  POKE  Z,81
     : GOTO  410
720 IF  A$= "-"  THEN  POKE Z,46
    : GOTO  410
730 IF  A$= "="  THEN  1600
740 IF  A$= "[CLEAR]"  THEN  370
750 IF  A$= "R"  THEN  1000
760 IF  A$= "D"  THEN  2000
770 GOTO  410
800 REM
810 CR=  FNB (0)
820 IF  A$= "N"  THEN  POKE  5327
    2,21 : GOTO  360
830 IF  A$= "E"  THEN  POKE 53272
    ,21 : F=0 : GOTO  1800
840 GOTO  410
1000 REM  OPZIONE PER REVISIONE
1010 POKE  53272,( PEEK (53272)  A
     ND  240)+12 : R=4 : C=0
1020 ZL=  FNA (0) : IL=32
1030 F=1 : PRINT  "[CLEAR]";
1040 PRINT  "[BIANCO]@ A B C D E F
      G" : PRINT
1050 PRINT  "H I J K L M N O" : PR
     INT
1060 PRINT  "P Q R S T U V W" : PR
     INT
1070 PRINT  "X Y Z [ / ] ^ _" : PR
     INT
1080 PRINT  "  ! " CHR$(34)" # $ %
      & '" : PRINT
```

```
1090 PRINT  "( ) * + , - . /" : PR
     INT
1100 PRINT  "0 1 2 3 4 5 6 7" : PR
     INT
1110 PRINT  "8 9 : ; < = > ?" : PR
     INT
1120 PRINT "[HOME]" SPC(25)"[RVS]O
     PZIONE[RVOFF]" : PRINT
1130 PRINT  SPC(22)"[RVS]N NUOVO C
     AR.[RVOFF]" : PRINT
1140 PRINT  SPC(22)"[RVS]E EDIT CA
     R.[RVOFF]" : PRINT
1150 PRINT  SPC(22)"[RVS]Q QUIT[RV
     OFF]" : PRINT
1155 PRINT  SPC(22)"[RVS][GIALLO]P
     REMI UN TASTO[RVOFF]" : PRINT

1156 PRINT  SPC(16)"[RVS][VERDE]US
     A 'CRSR' PER SCEGLIERE[RVOFF]
     " : PRINT
1157 PRINT  SPC(16)"         [RVS][V
     ERDE]CARATTERE[RVOFF]"
1160 BC= PEEK (55296)
1170 GOTO  410
1200 REM  OPZIONE DI EDIT
1210 PRINT "[HOME][BIANCO]" SPC(25
     )"[RVS]OPZIONE[RVOFF]" : PRIN
     T
1220 PRINT
1230 PRINT  SPC(P)"[RVS]+[RVOFF] D
     ISEGNA PUNTO" : PRINT
1240 PRINT  SPC(P)"[RVS]-[RVOFF] C
     ANCELLA" : PRINT
1250 PRINT  SPC(P)"[RVS]=[RVOFF] V
     ALORI CAR." : PRINT
1260 PRINT  SPC(P)"[RVS]R[RVOFF] R
     EVISIONE" : PRINT
1270 PRINT  SPC(P)"[RVS]Q[RVOFF] Q
     UIT" : PRINT
1280 PRINT  SPC(P)"[RVS]D[RVOFF] D
     ATI" : PRINT
1290 PRINT  SPC(22)" [RVS][GIALLO]
     PREMI UN TASTO[RVOFF]" : PRIN
     T
1295 PRINT  SPC(16)"[RVS][VERDE]US
     A 'CRSR' PER SCEGLIERE[RVOFF]
     " : PRINT
1297 PRINT  SPC(16)"         [RVS][VE
     RDE]CARATTERE[RVOFF]"
1300 RETURN
1500 REM  QUIT
```

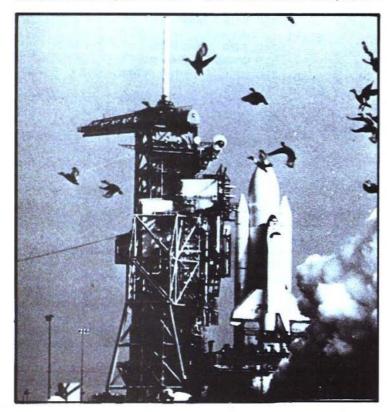
```
1520 POKE  53272,21
1530 POKE  53281,6 : POKE  53280,1
     4
1540 PRINT  "[CLEAR][CELESTE] CIAO
     !!"
1550 END
1600 REM  UPDATE
1610 PRINT  "[HOME]";
1620 X=CS+8*CR
1630 FOR  R=0  TO  7 : SM=0
1640 FOR  C=0  TO  7 : D=7-C
1650 SM=SM-2^D*( PEEK ( FNA(0) )=8
     1)
1660 NEXT  C
1670 POKE  X+R,SM
1680 PRINT  SPC (17) ; SM : PRINT
1690 NEXT  R : R=0 : C=0
1700 GOTO  410
1800 REM   EDIT
1810 PRINT  "[CLEAR]"
1820 X=CS+8*CR
1830 FOR  R=0  TO  7 : Y= PEEK (X+
     R)
1840 FOR  C=0  TO  7 : Z= FNA (0)
1850 Q=46 : Y=Y*2
1860 IF  Y >255  THEN  Q=81 : Y=Y-
     256
1870 POKE  Z,Q : POKE  Z+54272,1
1880 NEXT  C,R
1890 R=0 : C=0
1900 GOSUB  1200
1910 GOTO  410
2000 REM
2010 X=CS+8*CR
2020 PRINT  "[CLEAR][8 DOWN]"
2030 PRINT LN ; "DATA";
2040 PRINT   RIGHT$ ( STR$(X), LEN
      ( STR$(X))-1);
2050 FOR  I=X  TO  X+7
2060 PRINT  "," ;
2070 PRINT   RIGHT$ ( STR$( PEEK(I
     )), LEN ( STR$ ( PEEK(I)))-1)
     ;
2080 NEXT  I
2090 PRINT  : PRINT  "RUN [HOME]"
2100 POKE  828, PEEK (828) +1
2110 POKE  198,9
2120 FOR  I=0  TO  8
2130 POKE  I+631,13
2140 NEXT  I
2160 END
```

INPUT CONTROLLATO

di Ernesto Sidoti & Guido Minneci

Realizzando un programma in BASIC, spesso nasce la necessità di controllare i caratteri immessi durante un input. Necessità che diventa un obbligo quando il software realizzato verrà utilizzato da diverse persone, magari a scopi professionali. Un'efficace routine di controllo dell'input non solo limita le possibilità di errore da parte dell'utente, ma rende anche il programma più elegante e scorrevole alla lettura. Ciò si traduce in un grande risparmio di tempo durante la fase di correzione e modifica. È logico che per programmi molto piccoli o che richiedono pochi dati in ingresso da tastiera, una routine di controllo dell'input potrebbe risultare superflua. Ciò non accade, ad esempio, se il programma è di carattere gestionale, quindi di dimensioni piuttosto elevate.

La routine riportata può essere considerata quale programma da inserire opportunamente nel vostro programma principale. Per motivi di ordine è consigliabile posizionare il listato proposto in coda al programma, e richiamare la funzione quando serve con l'istruzione GOSUB. È chiaro che in questo caso si deve eseguire una opportuna rinumerazione della routine stessa in funzione della posizione che prenderà nell'ambito del programma.

Il programma che troverete nell'ambito di questo articolo è composto da un piccolo programma-prova per mostrare le capacità della routine e le modalità d'uso, quindi segue la routine vera e propria. Ricordate che prima di eseguire l'input bisogna definire il posto dove si vuole che lampeggi il cursore. Ciò si ottiene fissando il numero della colonna e della riga rispettivamente con le variabili X% e Y%. Bisogna inoltre fissare la variabile 1% per definire la lunghezza dell'input e la variabile T% per definire il tipo, che può essere alfanumerico, cioè accettare lettere e numeri, o numerico. Definite, queste variabili, si manda il programma ad eseguire la routine. Al rientro il dato immesso sarà associato alla variabile X$ pronto per essere archiviato in un'altra variabile. Fatta quest'ultima operazione, non resta che azzerare la variabilità X$ e passare al successivo input.

Breve descrizione del programma prova

100 definisce il volume della nota che suona quando si fa un'operazione illecita.

190 il ciclo che definisce il colore del cursore.

210-350 definisce i colori di fondo schermo e disegna una maschera.

410 definisce le variabili X%, Y%, L%, T%, rispettivamente numero della colonna, della riga, della lunghezza massima dell'input e infine del tipo (o alfanumerico, 1 numerico).

420-580 Esempio di riempimento dàti sulla maschera tramite l'input controllato.

Quando si sarà dato il comando RUN al programma verrà disegnata la maschera di input, e quindi si entra proprio nella routine in oggetto. Se tentate ad esempio di spostarvi con i cursori (CRSR / up / down), suonerà una nota per avvisare che si sta premendo un tasto illecito. Il suono si avvertirà anche premendo i caratteri grafici o i tasti di funzionamento (F1 F2 F3 ecc.). Se durante l'input viene premuta una lettera errata, o comunque si volesse correggere il dato in fase di inserimento, è sufficiente spostarsi con i tasti (CRSR/left/right) ed eseguire la o le correzioni. Se lo spostamento supera la lunghezza fissata per quel dato allora sarà segnalato l'errore. Se tutta la stringa di input è errata o se si vuole azzerare l'introduzione in corso è sufficiente cancellare con un solo tasto, quello di CLR (CLR/HOME).

L'input controllato inizia alla riga 800 e termina alla 1150, mentre dalla 1160 alla 1190 si genera un segnale acustico per avvisare l'operatore di eventuali errori.

```
10 REM  ******************
20 REM  * PROGRAMMA PROVA *
30 REM  ******************
40 :
50 REM  ---------------------
60 REM  - IMPOSTA IL VOLUME -
70 REM  - DELLA NOTA DI     -
80 REM  - AVVERTIMENTO      -
90 REM  ---------------------
100 POKE 54296,15
110 :
120 PRINT"[CLEAR]"
130 :
140 REM -----------------------
150 REM - DEFINISCO IL COLORE -
160 REM - DEL CURSORE         -
170 REM -----------------------
180 :
190 FOR A=55296 TO 56295:POKE A,1:NEXT
200 :
210 POKE 53280,0:POKE 53281,0:A=3:DIM A$(1,6):B=1
220 PRINT"[CLEAR][BIANCO][7 RIGHT]┌──────────────────────┐
230 PRINT"[7 RIGHT]|     PROGRAMMA PROVA    |
240 PRINT"[7 RIGHT]└──────────────────────┘
250 PRINT TAB(2)"[2 DOWN]┌──────────────────────┐
260 FOR S=1 TO 16
270 PRINT TAB(2)"|[35 RIGHT]|"
280 NEXT
290 PRINT TAB(2)"└──────────────────────┘"
300 PRINT"[HOME]":PRINT TAB(A)"[7 DOWN]NOME      ........................"
310 PRINT TAB(A)"[DOWN]COGNOME      ........................"
320 PRINT TAB(A)"[DOWN]INDIRIZZO ........................"
330 PRINT TAB(A)"[DOWN]TELEFONO      ........................"
340 PRINT TAB(A)"[DOWN]COMUNE      ........................"
350 PRINT TAB(A)"[DOWN]ACCONTO      ............"
360 REM  **********************
370 REM  *                    *
380 REM  *  RICHIAMO ROUTINE  *
390 REM  *                    *
400 REM  **********************
410 X%=13:L%=24:T%=0:Y%=7
420 FOR A=1 TO 5
430 Y%=Y%+2
440 GOSUB 800
450 A$(B,A)=X$:X$=""
460 NEXT
470 X%=12:Y%=19:L%=14:T%=1
480 GOSUB 800
490 A$(B,6)=X$:X$=""
500 REM  **********************
```

```
510 PRINT"[HOME]"
520 FOR S=0 TO 19:PRINT:NEXT
530 PRINT TAB(9)"[RVS]FINE INPUT[RV
    OFF]"
540 GET D$:IF D$="" THEN 540
550 PRINT"[CLEAR]"
560 FOR A=1 TO 6
570 PRINT TAB(4)A$(1,A):PRINT
580 NEXT:END
590 :
600 :
610 REM ***************************
620 REM *                         *
630 REM *    INPUT CONTROLLATO    *
640 REM *                         *
650 REM *           DI            *
660 REM *                         *
670 REM *    ERNESTO SIDOTI  E    *
680 REM *                         *
690 REM *      GUIDO MINNECI      *
700 REM *                         *
710 REM ***************************
720 :
730 REM  ---------------------------
740 REM  X%..POSIZ. X SUL VIDEO
750 REM  Y%..POSIZ. Y SUL VIDEO
760 REM  L%..LUNGHEZZA DELL' INPUT
770 REM  T%..TIPO (0.ALFA/1.NUM)
775 REM  X$..VARIABILE DI OUTPUT
780 REM  ---------------------------
790 :
800 PRINT"[HOME]";:FOR AA=1 TO Y%-1
    :PRINT"[DOWN]";:NEXT:I%=0:IF Y%
    =1 THEN PRINT"[UP]";
810 FOR AA=1 TO X%:PRINT"[RIGHT]";:
    NEXT:FOR AA=1 TO L%:PRINT" ";:N
    EXT:FOR AA=2 TO L%
820 PRINT"[LEFT]";:NEXT:IF T%=1 THE
    N C$=LEFT$(X$,1):IF C$="" OR VA
    L(C$)<>0 THEN C$="+"
830 IF T%=1 THEN X$=C$+RIGHT$(STR$(
    VAL(X$)),LEN(STR$(VAL(X$)))-1):
    I%=1
840 IF T%=1 THEN PRINT"[LEFT]";
850 PRINTX$;:IF T%=1 THEN PRINT"[RI
    GHT]";
860 FOR AA=1 TO LEN(X$):PRINT"[LEFT
    ]";:NEXT
870 GET C$:IF C$<>"" AND FL=0 THEN
    940
880 IF C$="[HOME]" THEN 800
890 K%=PEEK(1024+(Y%-1)*40+X%+I%)
900 IF K%>0 AND K%<127 THEN POKE 10
    24+(Y%-1)*40+X%+I%,K%+128:FL=1
910 IF K%>127 AND K%<255 THEN POKE
    1024+(Y%-1)*40+X%+I%,K%-128:FL=
    0
920 IF C$="" THEN 870
930 IF C$<>"" AND FL=1 THEN 910
940 IF C$="[HOME]" THEN 800
950 C%=ASC(C$):IF C%=145 THEN GOSUB
     1160:GOTO 870
960 IF C%=13 THEN 1100
970 IF T%=1 AND I%=0 THEN IF C%=43
    OR C%=45 THEN 1000
980 IF T%=1 AND (C%=29 OR C%=157) T
    HEN 1000
990 IF T%=1 AND (C%<48 OR C%>57) TH
    EN GOSUB 1160:GOTO 870
1000 IF C%=17 THEN GOSUB 1160:GOTO 8
     70
1010 IF C%=157 AND I%<=0 THEN GOSUB
     1160:GOTO 870
1020 IF C%=157 AND I%>0 THEN I%=I%-1
     :PRINT"[LEFT]";:GOTO 870
1030 IF C%=29 AND I%>=L%-1 THEN GOSU
     B 1160:GOTO 870
1040 IF C%=29 AND I%<L% THEN I%=I%+1
     :PRINT"[RIGHT]";:GOTO 870
1050 IF T%=1 AND I%<>1 AND (C%=43 OR
      C%=45) THEN 1080
1060 IF T%=1 AND (C%<48 AND C%>57) T
     HEN GOSUB 1160:GOTO 870
1070 IF C%<32 OR C%>93 THEN GOSUB 11
     60:GOTO 870
1080 IF I%>L%-1 THEN GOSUB 1160:GOTO
      870
1090 I%=I%+1:PRINTC$;:GOTO 870
1100 X$="":IF T%=1 THEN FOR AA=0 TO
     L%-1:C=PEEK(1024+(Y%-1)*40+X%+A
     A):X$=X$+CHR$(C)
1110 IF T%=1 THEN NEXT:GOTO 1150
1120 FOR AA=0 TO L%-1:C=PEEK(1024+(Y
     %-1)*40+X%+AA)
1130 IF  C>27  THEN  C=C-64
1140 X$=X$+CHR$(C+64):NEXT
1150 RETURN
1160 REM  ******* SOUND ********
1170 POKE 54277,8:POKE 54278,17:POKE
      54273,89:POKE 54272,15:POKE 54
     274,0
1180 POKE 54275,9:POKE 54276,65
1190 FOR TT=1 TO 90:NEXT:POKE 54276,
     0:RETURN
```

LA MAPPA DEI QUATTRO COLORI

di **Eugenio Coppari**

"Un gioco matematico ben fatto informa più di una dozzina di articoli mediocri". Questa frase, pronunciata dal matematico inglese Littlewood, costituisce un approccio originale, seppur indiretto, al ramo matematico oggetto di analisi in questo articolo: la Topologia, materia che si propone di studiare le proprietà invariabili delle figure geometriche sottoposte ad una distorsione.

Immaginiamo di considerare una parte della nostra penisola costituita da 3 regioni tra loro confinanti, ad esempio Lombardia, Trentino Alto Adige e Veneto.

Ora immaginiamo che un qualche evento naturale altéri la forma di questi 3 regioni senza peraltro variare i confini. In queste condizioni la proprietà di cui godono le 3 regioni, essere confinanti, non viene alterata dal fenomeno naturale che ha coinvolto queste aree, risulta quindi ragiovevole da un punto di vista geometrico considerare questa proprietà come topologica.

Questa breve premessa sul concetto di topologia consente di introdurre l'argomento su cui è incentrato il programma: il teorema dei quattro colori. In matematica esistono dei teoremi il cui significato è facilmente comprensibile dalla maggioranza delle persone, ma che nonostante questo hanno occupato intere generazioni di matematici per renderne possibile la dimostrazione oppure la confutazione. Il teorema dei quattro colori ne è un classico esempio. Tra il periodo in cui è stato formulato il problema e quello in cui lo si è risolto sono trascorsi più di 120 anni e la vicenda umana delle parecchie persone che si sono impegnate nella sua analisi ha finito per diventare la storia di questo teorema.

Analizziamone l'enunciato: "Per ogni suddivisione del piano in regioni che non si sovrappongano, è sempre possibile colorare tali regioni in maniera tale che due adiacenti non abbiano lo stesso colore". La formulazione di questo teorema risale a oltre 100 anni addietro ed è, con ogni probabilità, da addebitare a uno studente universitario inglese del secolo scorso. La situazione prospettata dall'enunciato del teorema indusse i matematici di quel periodo a ritenere che la soluzione dell'enigma dovesse essere relativamente semplice grazie anche ai notevoli mezzi teorici di cui la matematica era già allora in possesso. Un classico esempio di questa distorsione iniziale dei veri connotati del problema è rappresentato dal tentativo del matematico De Morgan che ritenne di avere dimostrato il teorema con una proposizione da lui ritenuta erroneamente analoga. Egli riuscì in effetti a verificare che all'interno di qualsiasi mappa non possono esistere 5 regioni mutuamente adiacenti, ma ne trasse una conclusione errata ritenendo di avere risolto l'enigma precedentemente citato. Un breve attimo di riflessione è sufficiente per comprendere che le 2 proposizioni non sono assolutamente equivalenti.

Colui che è riuscito a fornire un contributo decisivo, già nel secolo scorso, alla soluzione di questo problema è stato, ironia della sorte, un "dilettante" delle discipline matematiche: l'avvocato Arthur Kempe. Come spesso è accaduto nella dimostrazione delle maggiori proposizioni matematiche, anche Kempe ha operato un tentativo di dimostrazione per assurdo ("Reductio Absurdum"). Cercheremo di spiegarlo brevemente anticipando inizialmente alcune importanti definizioni che ci serviranno nel seguito: una carta (mappa) si dice normale quando in ciascuno dei vertici che la costituiscono non convergono più di 3 spigoli e inoltre ogni regione non si trova mai completamente racchiusa all'interno di un'altra delle confinanti. Kempe assunse come postulato l'esistenza di una carta pentacromatica, cioè una mappa per la quale non sono sufficienti 4 colori al fine di completarne la tinteggiatura.

La speranza che una asserzione di questo genere potesse condurre a delle contraddizioni di carattere logico ha permesso, in tempi più recenti, di giungere alla dimostrazione del teorema.

Kempe dimostrò innanzitutto che se esisteva una mappa che necessitava di 5 colori, allora doveva esistere almeno una carta normale

COMPUTER QUESTO MESE È QUESTO

COMPUTER

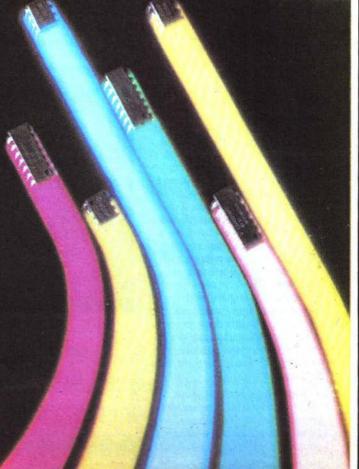
N.68 - lire 3000 il "newsmagazine" dell'informatica

Unix, la terra promessa

L'erede di Pascal

Il futuro in un disco

Verso la fabbrica robotizzata

Microprocessori, otto grandi a confronto

pentracromatica per la quale 4 colori non sono sufficienti. Egli verificò inoltre, in sintonia con le condizioni poste inizialmente, l'esistenza di una carta pentacromatica minimale, il che significava che ogni altra mappa costituita da un numero minore di regioni può essere colorata con l'ausilio di soli 4 colori. E poiché all'interno di ogni mappa sono naturalmente possibili svariate configurazioni di regioni, Kempe riuscì ingegnosamente a dimostrare che alcune di esse sono inevitabili. Il concetto appena espresso evidenzia l'esistenza di un insieme, composto da un numero finito di configurazioni, strutturato in modo tale che almeno una di esse sia presente in qualsiasi carta da noi costruita.

Questo aggregato è composto da 4 elementi: 1 regione con 2 confinanti, 1 regione con 3 confinanti, 1 regione con 4 confinanti, 1 regione con 5 confinanti. Una breve verifica con carta e matita dovrebbe essere sufficiente a convincerci della bontà di tale affermazione. Ora dimostrando l'impossibilità da parte di uno degli elementi dell'insieme di appartenere ad una carta pentacromatica minimale si giungerebbe alla dimostrazione della congettura dei 4 colori. Kempe riuscì a dimostrare questa incompatibilità in riferimento ai primi 3 elementi dell'insieme, ma fallì quando si trattò di verificare che una regione con 4 confinanti non può appartenere ad una carta pentacromatica minimale. Dovrebbe risultare sufficientemente chiaro che Kempe nel suo tentativo di dimostrazione ha cercato con successivi passaggi di diminuire l'area soggetta ad analisi al fine di semplificare la dimostrazione del teorema.

Questo fallimento, seppure relativo, cominciò a far dubitare che si potesse giungere ad una dimostrazione della congettura. Soprattutto quando all'inizio del novecento le idee di indecidibilità e incompletezza propugnate dal logico austriaco Gödel ponevano in evidenza il fatto che esistessero dei teoremi che non erano né dimostrabili né confutabili.

Il programma si basa sulle esperienze dei veri solutori del problema che fanno a capo a Heesch. Il gruppo di solutori sono partiti da dove Kempe aveva fallito. Trovarono cioè una soluzione più valida che consentiva e consente tutt'ora di affermare che è possibile una risoluzione della congettura, pur tuttavia non specificandone l'esatto algoritmo.

Il listato che vi proponiamo sviluppa il concetto principale del gruppo Heesch che dice:

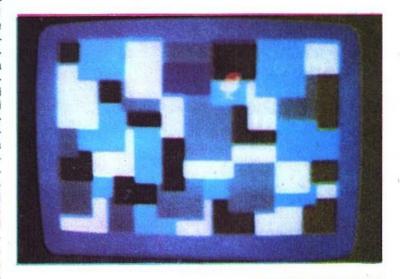
Prendendo una regione a caso della mappa, coloratela con il primo colore. Avvalendoci sempre del primo colore tinteggiamo tutte le zone della mappa non adiacenti tra loro. Quando questa operazione non sarà più possibile ricorriamo al secondo colore, ripetiamo la procedura precedente non tinteggiando le zone già colorate.

Giunti al quarto colore si potranno verificare due eventualità:

1 - Quattro coloro sono stati sufficienti per colorare tutta la mappa.

2 - Al termine dei quattro colori esistono ancora delle zone adiacenti non colorabili senza contraddire il principio della congettura.

Sarà quindi necessario azzerare la nostra mappa e iniziare nuovamente la procedura descritta in precedenza. Questa operazione dovrà essere continuata sino a che non si otterranno i risultati evidenziati al punto 1.

```
1000 REM *************************
1010 REM *                       *
1020 REM * TEOREMA DEI 4 COLORI  *
1030 REM *                       *
1040 REM *   DI EUGENIO COPPARI  *
1050 REM *                       *
1060 REM *************************
1070 PRINT"[CLEAR][7 DOWN]
```

Computer questo mese é questo...

E QUESTO ....

```
        [RVS]                    [RVOFF]"
1080 PRINT "                  [RVS] IL T
     EOREMA [RVOFF]"
1090 PRINT"                   [RVS]
          [RVOFF]
1100 PRINT "[DOWN]
     [RVS]      [RVOFF]
1110 PRINT "                       [RVS]
     DEI [RVOFF]
1120 PRINT "                       [RVS]
        [RVOFF]
1130 PRINT"[DOWN]               [RVS]
                    [RVOFF]"
1140 PRINT"               [RVS] QUATTRO
      COLORI [RVOFF]"
1150 PRINT"               [RVS]
             [RVOFF]"
1160 FOR UB=0 TO 3
1170 READ C
1180 FOR YI=1 TO 3
1190 POKE 1837+UB*(3)+YI,160:POKE 56
     109+UB*(3)+YI,C
1200 POKE 1877+UB*(3)+YI,160:POKE 56
     149+UB*(3)+YI,C
1210 POKE 1917+UB*(3)+YI,160:POKE 56
     189+UB*(3)+YI,C
1220 NEXTYI
1230 NEXTUB
1240 RESTORE
1250 FOR II=1 TO 10000:NEXT
1260 REM ***********************
1270 REM * CREAZIONE DELLA MAPPA*
1280 REM *   ELEMENTI  PER LA   *
1290 REM *  VALUTAZIONE  DEI    *
1300 REM *       CONFINI        *
1310 REM ***********************
1320 DIM O(60,80):DIM TL(80):DIM XX(
     60):DIM B(60):DIM J(1080):DIM D
     R(60)
1330 PRINTCHR$(147):K=1
1340 C=54272:Y1=1023+Q*(40)
1350 IF Z=40 THEN 2330
1360 A=INT(RND(1)*5)+3
1370 IF A=CL(1) THEN 1360
1380 CL(1)=A
1390 IF Q=10 THEN Z1=25
1400 IF Z1=25 THEN 1470
1410 A1=INT(RND(1)*5)+3+Q
1420 IF A1=CL(2) THEN 1410
1430 FOR DD=0 TO A-1
1440 IF PEEK((Y1+1+DD+(40*(A1-1)))-4
     0)<>32 THEN 1360
1450 NEXTDD
1460 Z1=A1+Q
1470 Z=Z+A
1480 IF Z1>23 THEN Z1=25:A1=Z1-Q
1490 IF 40-Z=2 OR 40-Z=1 THEN Z=Z-A:
     A=40-Z:Z=40
1500 IF Z>40 THEN Z=Z-A:GOTO 1360
1510 CL(2)=A1
1520 CL(3)=Z
1530 G=A-2
1540 G1=A1-2
1550 IF PEEK(Y1+1)<>32 THEN 1620
1560 R%=R%+1
1570 POKE Y1+1,79
1580 POKE C+Y1+1,1
1590 POKE 49151+R%,79
1600 O(K,R%)=Y1+1
1610 J(Y1+1-983)=K
1620 IF PEEK(Y1+1+(40*(A1-1)))<>32 T
     HEN 1690
1630 R%=R%+1
1640 POKE Y1+1+(40*(A1-1)),76
1650 POKE C+Y1+1+(40*(A1-1)),1
1660 POKE 49151+R%,76
1670 O(K,R%)=(Y1+1+(40*(A1-1)))
1680 J(Y1+1+(40*(A1-1))-983)=K
1690 IF G<>0 THEN GOSUB 1900
1700 Y1=1023+A+X+Q*(40)
1710 IF PEEK(Y1)<>32 THEN 1780
1720 R%=R%+1
1730 POKE Y1,80
1740 POKE C+Y1,1
1750 POKE 49151+R%,80
1760 O(K,R%)=Y1
1770 J(Y1-983)=K
1780 IF PEEK(Y1+(40*(A1-1)))<>32 THE
     N 1850
1790 R%=R%+1
1800 POKE Y1+(40*(A1-1)),122
1810 POKE C+Y1+(40*(A1-1)),1
1820 POKE 49151+R%,122
1830 O(K,R%)=(Y1+(40*(A1-1)))
1840 J(Y1+(40*(A1-1))-983)=K
1850 IF G1<>0 THEN GOSUB 2060
1860 X=X+A
1870 XX(K)=R%
1880 IF (A>2) AND (A1>2) THEN GOSUB
     2220
```

```
1890 TL(K)=R%:K=K+1:CC=CC+K:R%=0:GOT
     O 1350
1900 FOR DD=1 TO G
1910 IF PEEK(Y1+1+DD)<>32 THEN 1980
1920 R%=R%+1
1930 POKE Y1+1+DD,119
1940 POKE C+Y1+1+DD,1
1950 POKE 49151+R%,119
1960 O(K,R%)=Y1+1+DD
1970 J(Y1+1+DD-983)=K
1980 IF PEEK(Y1+1+DD+(40*(A1-1)))<>3
     2 THEN 2050
1990 R%=R%+1
2000 POKE Y1+1+DD+(40*(A1-1)),111
2010 POKE C+Y1+1+DD+(40*(A1-1)),1
2020 POKE 49151+R%,111
2030 O(K,R%)=(Y1+1+DD+(40*(A1-1)))
2040 J(Y1+1+DD+(40*(A1-1))-983)=K
2050 NEXT:RETURN
2060 FOR TT=1 TO G1
2070 IF PEEK(Y1-(A-1)+TT*40)<>32 THE
     N 2140
2080 R%=R%+1
2090 POKE Y1-(A-1)+(TT*40),116
2100 POKE C+Y1-(A-1)+(TT*40),1
2110 POKE 49151+R%,116
2120 O(K,R%)=(Y1-(A-1)+(TT*40))
2130 J(Y1-(A-1)+(TT*40)-983)=K
2140 IF PEEK(Y1+TT*40)<>32 THEN 2210
2150 R%=R%+1
2160 POKE Y1+(TT*40),106
2170 POKE C+Y1+(TT*40),1
2180 POKE 49151+R%,106
2190 O(K,R%)=(Y1+(TT*40))
2200 J(Y1+(TT*40)-983)=K
2210 NEXTTT:RETURN
2220 FOR PA=1 TO (A1-2)
2230 FOR PB=(A-2) TO 1 STEP -1
2240 IF PEEK(Y1+40*PA-PB)<>32 THEN 2
     300
2250 R%=R%+1
2260 POKE Y1+40*PA-PB,81
2270 POKE  49151+R%,81
2280 O(K,R%)=(Y1+40*PA-PB)
2290 J(Y1+40*PA-PB-983)=K
2300 NEXTPB
2310 NEXTPA
2320 RETURN
2330 R%=0:IF Z1=25 THEN K=K-1:GOTO 2
     430
2340 Q=Q+2:Z=0:X=0
2350 GOTO  1340
2360 REM ***********************
2370 REM * VALUTAZIONE ELEMENTI *
2380 REM *       AQUISITI       *
2390 REM *   PRECEDENTEMENTE    *
2400 REM *     E CONSEGUENTE    *
2410 REM *  COLORAZIONE  MAPPA  *
2420 REM ***********************
2430 E=1024:I=2023:GOTO 2690
2440 QQ=INT(RND(1)*K)+1
2450 FOR DE=1 TO M2
2460 IF QQ=B(DE) THEN 2440
2470 NEXT
2480 G3=G3+1:DR(G3)=QQ
2490 FOR CC=1 TO TL(QQ)
2500 POKE O(QQ,CC),160:POKE 55296+(O
     (QQ,CC)-1024),C
2510 NEXT
2520 FOR FF=1 TO XX(QQ)
2530 N=O(QQ,FF)
2540 QO=J(N+40-983):GOSUB 2590
2550 QO=J(N-40-983):GOSUB 2590
2560 IF (N-1063)/40<>INT((N-1063)/40
     ) THEN QO=J(N+1-983):GOSUB 2590
2570 IF (N-1024)/40<>INT((N-1024)/40
     ) THEN QO=J(N-1-983):GOSUB 2590
2580 NEXTFF:GOTO 2440
2590 FOR VV=1 TO M2
2600 IF B(VV)=QO THEN RETURN
2610 NEXTVV
2620 IF QO=0 THEN RETURN
2630 M2=M2+1:IF M2=K THEN 2650
2640 B(M2)=QO:RETURN
2650 FOR ES=1 TO G3
2660 B(ES)=DR(ES)
2670 NEXT
2680 M2=G3
2690 KU=KU+1
2700 IF KU>5 THEN 2730
2710 READ  C:GOTO 2440
2720 DATA  1,0,12,3,6
2730 GOTO 2730
```

ANNUNCI

Vendo. Programmi per Commodore 64, dispongo delle ultime novità di mercato: Clon Machine; Disc Key, Unguard, per proteggere e sproteggere. Informer, gestionali vari, stress, ingegneria, condominio e gli ultimi giochi del 1984. Richiedere lista. (Augusto Bernardini - Via Valle Verde n° 5 - 05100 Terni)

Vendo / Scambio. Programmi per Commodore e VIC 20. Telefonare o spedire lettere con la lista dei propri programmi per fare scambio con i miei a: Luca Lauro - Via Vittoria Colonna 220 - 80071 Ischia (Napoli) tel. 081 / 992589.

Vendo / Cambio. Programmi su disco o cassetta per Commodore 64. Vastissimo assortimento giochi e utilità prezzi fallimentari. Scrivere. Attenzione annuncio sempre valido (Giorgio Zoli - Via Alberto Mario n° 27 - Forlì)

Vendo / Cambio Software per Commodore C64 Giochi e utility. Inviate vostre liste (Leoncini Massimo - Casella Postale 85 - 16035 Rapallo (GE)

Vendo Nastro con 10 videogames ciascuno scelti su lista che che invierò gratis a richiesta. L. 50.000 a cassetta. Caricamento ultraveloce con apposito programma registrato in testa. (Marco Polenta - Via Baccarani n° 7 - 60100 Ancona - Tel. 071 / 55089 ore pasti)

Vendo Programmi su cassetta per ZX Spectrum e Commodore 64. Aspetto le vostre telefonate! (Pisono Fabio - Via Olcella 102 - 20022 Busto Garolfo - Milano - Tel. 0331 / 568337)

Vendo per CBM 64 **Giochi** in L.M.. Sei programmi a scelta 40.000 lire, escluse spese di spedizione. Oltre 100 programmi per tutti i gusti: spaziali, avventure in 3D, d'azione, guerra, sport, utility, ecc. In omaggio per ogni ordine programma per bioritmi grafici a colori. (Massimo Selleri - Via Don Minzoni 72 - 45100 Rovigo)

Vendo Cassetta del gioco PAC-MAN a L. 6.000 oppure scambio CPN cassetta differente. Annuncio sempre valido. Solo Milano e provincia. (Guido Gafforelli - Via Settala n° 1 - 20124 Milano)

Vendo / Compro Programmi Vic 20 o cedo programmi per espansioni di memoria. (Alberto Zalla - Via Gasparinetti 22 - 31100 Treviso - Tel. 0422 / 562447)

Vendo / Cambio / Compro per Commodore **Programmi** di ogni genere. Rispondo a tutti. (Matteo Doveri - Via I° Maggio 15 - 56025 Pontedera (Pisa) Tel. 0587 / 5212154)

Vendo / Cambio Software Commodore 64 Giochi Utility ampio elenco di programmi. Telefonare o scrivere. Rispondo a tutti. (Roberto Quaglia - Via Martinazzoli 2 - 20161 Milano - Tel. 02 / 6462130 ore 20-22)

Vendo circa **300 Programmi** per C64, dai videogames eccezionali alle più svariate utility. I prezzi sono ultra stracciati. Richiedere la lista o telefonare. (Silvio Panetta - Via Matilde Serao 4 - 10141 Torino - Tel. 011 / 373647 ore 19-20,30)

Vendo o meglio **Cambio** tutti i tipi **Software** per Commodore 64. Cerco anche amici per scambio opinioni sul linguaggio macchina del 6510. (Francesco Billari - Via Ciovasso 4 - 20121 Milano - Tel. 02 / 870951 ore pasti)

Vendo / Cambio circa **600 Programmi** per CBM 64 (Luca Paradisi - Via degli Ebrei 4 - 57100 Livorno - Tel. 0586 / 801834 ore 14-15-20-21)

Vendo Programmi VIC 20, cassetta con 60 programmi scelti tra una vasta gamma di 200 programmi tra cui giochi e utility a L. 30.000 + spese postali (Luciano Baglioni - Via della Verna 20 - 00141 Roma - Tel. 06 / 899286 ore pasti)

Cambio / Vendo Programmi e **Giochi** per CBM 64 solo su nastro registrato. Dispongo di oltre 300 programmi. Inviatemi la vostra lista e io vi spedirò la mia. Rispondo a tutti gli interessati. Annucio sempre valido? (Claudio Bacci - Via dei Pelaghi 165 - 57100 Livorno - Tel. 0586 / 852551 ore 9-12 / 16-19)

Vendo Programmi gioco e gestionali per Commodore 64 e VIC 20 - Telefonare ore pasti o scrivere. (Gianni Caudino - Via Graglia 18 - 10136 Torino - Tel. 011 / 352830 ore 13-14 / 19-22)

Vendo per CBM 64 a L. 38.000 trattabili **Cartuccia "jupiter Lander"**. Sono pure disposto a scambiarla con altra cartridge per CBM 64 (giochi, utility). Rispondo celermente a tutti. (Addabbo Giovanni - Via a Monte Vettore 46 - 60131 Ancona)

Vendo CBM 64 - Sharp 700 - CBM 8000, qualsiasi programma; per 64 Simon's Basic, 40. Compilatori 80 colonne L. 30.000, Sintetizzatore vocale velocizzatore nastro word processing, data base, lastone. Prezzi bassi (Giovanelli Claudio - Via Ripamonti 195 Milano - Tel. 02 / 563105)

Vendo / Cambio oltre **500 Programmi** per Commodore 64 tra i quali molti di utilità. (Loredana Colò - Via Goti 99 - 42019 Scandiano (RE) - Tel. 0522 / 855130 ore pasti)

Vendo / Cambio Programmi per Commodore 64. Dispongo di circa 400 titoli a ottimi prezzi. (Fabrizio Aggigoni - Via Sem Benelli 2 - 20151 Milano - Tel. 02 / 3083592 ore dopo le 20,30)

Vendo / Cambio Programmi per il VIC 20 e 64. Inviare L. 500 per la risposta e lista, cerco espansione 16K in buone condizioni max. 35.000 lire (Raineri Antonio - Via Mazzini 1 - 20077 Melegnano - Milano)

Vendo per CBM 64 **Favolosi programmi**, giochi utility, didatici a prezzi stracciatissimi. Richiedere elenco allegato un francobollo da 400 lire. (Franco Foschi - Via Corelli 24 - 47100 Forlì - Tel. 0543 / 65384 ore 14-19)

Vendo Commodore VIC 20 + registratore + espansione 16 Krem + vari videogiochi e vari programmi. Tutto a L. 450.000 (Mario Lombardi - Via Palmanova 209 - 20132 Milano - Tel. 02 / 2567039 ore pasti)

Vendo registratore dedicato + CBM 64 + 2 joisticks + stampante MPS 801 + 2000 fogli M.C. + monitor fosofori verdi + plotter 1520: anche a pezzi singoli. Prezzi: 100.000 + 640.000 + (290.000 × 2) + 560.000 + 42.000 + 250.000 + 415.000. Vendo causa trasferimento dall'Italia. Tutti i pezzi sono nuovi. (Alberto Cicale - Via C. Porta, 24 - 28100 Novara - Tel. 0321 / 29805)

Vendo Vic 20 completo di manuale istruzioni L. 178.000 con programmi in omaggio espansione 16K L. 120.000 - Cartridge HES Writer con manuale L. 70.000. (Durello Antonio - Via Durer 31 - 35100 Padova - Tel. 049 / 612435 ore pasti)

Compro o scambio Programmi matematici statistici - gestionali e utility per Commodore 64. Sono interessato anche a giochi solo se molto belli (H.R. and L.M.). (Averna Giuseppe - Via Dei Mille 73 - 27029 Vigevano Pavia)

Gruppo utenti CBM 64 scambia / vende / compra software. Invia la tua lista. (Enzo Petrizzo - Via Nazionale 157 - 84030 Padula Scalo - Tel. 0975 / 74022)

Compro usato **Commodore C64** completo di registratore, sempre questo rientri nelle mie capacità economiche, avrei preferito altrimenti usufruire di vistri nuovi prodotti! (Veneri Marco - Via P.F. Calvi 12 - 37100 Verona - Tel. 045 / 583937 ore 14)

Se vuoi abbonarti

Registrate il mio abbonamento annuale a Commodore.

☐ Ho versato oggi stesso il canone di Lire 25.000 a mezzo c/c postale nº 31532203 intestato a: Commodore Systems Editoriale - V.le Famagosta, 75 - 20142 Milano

☐ Accludo assegno per lire 25.000 banca .... nº .... a favore di Commodore Systems Editoriale

Il mio computer è: VIC 20 ☐, C 64 ☐, altro (specificare) ....

Ho ☐ / non ho ☐ la stampante, ma voglio ☐ comprarla.

Preferisco programmi di gioco ☐, didattici ☐, d'utilità ☐, altro ....

Nome .... Cognome ....

Via .... nº .... CAP. ☐☐☐☐☐ Città ....

Tel. ....

Se vuoi collaborare

Registratemi fra i collaboratori regolari di Commodore.

A titolo di prova vi invio un articolo e la cassetta col programma "........ ...." di cui vi garantisco l'assoluta originalità autorizzandovene la pubblicazione.

☐ Scrivetemi all'indirizzo sottoindicato ....

Nome ....

Via .... Nº ....

Tel. .... CAP .... Città ....

Se vuoi un consiglio o consigliarci

HELP ....

Nome ....

Via .... nº .... CAP. ☐☐☐☐☐ Città ....

Tel. .... Orario ....

Il mio computer è configurato:

			Sono in possesso	No	Ho intenzione di acquistare
Vic 20 ☐	espanso a ....K		☐	☐	☐
C 64 ☐			☐	☐	☐
Floppy ☐	quale: 1541 ☐	altro	☐	☐	☐
Stampante ☐	quale: MPS801 ☐	altro	☐	☐	☐
Plotter ☐	quale: 1520 ☐	altro	☐	☐	☐
Registratore ☐	quale: 1530 ☐	altro	☐	☐	☐

Televisore ☐, TV-Monitor ☐, Monitor ☐, Colore ☐, B/N ☐

Nome .... Cognome ....

Via .... nº .... CAP. ☐☐☐☐☐ Città ....

Tel. ....

Se vuoi vendere o comprare

Vendo ☐ Compro ☐ ....

Nome ....

Via .... nº .... CAP. ☐☐☐☐☐ Città ....

Tel. .... Orario ....

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

v.le Famagosta, 75
20142 Milano

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

v.le Famagosta, 75
20142 Milano

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

v.le Famagosta, 75
20142 Milano

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

v.le Famagosta, 75
20142 Milano

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

v.le Famagosta, 75
20142 Milano

E l'ultimo fascicolo l'hai già comperato?